

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2002-0070785
Application Number

출원년월일 : 2002년 11월 14일
Date of Application NOV 14, 2002

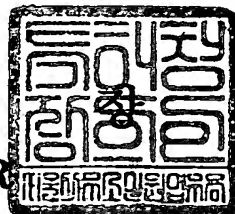
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 06 월 13 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2002. 11. 14
【발명의 명칭】	플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법
【발명의 영문명칭】	FLASH MEMORY, ACCESS APPARATUS AND METHOD USING FLASH MEMORY
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	김동진
【대리인코드】	9-1999-000041-4
【포괄위임등록번호】	2002-007585-8
【발명자】	
【성명의 국문표기】	김범수
【성명의 영문표기】	KIM, Bum Soo
【주민등록번호】	690121-1019710
【우편번호】	431-080
【주소】	경기도 안양시 동안구 호계동 1055-1 무궁화아파트 703동 203호
【국적】	KR
【발명자】	
【성명의 국문표기】	정재욱
【성명의 영문표기】	CHEONG, Jae Wook
【주민등록번호】	740415-1537014
【우편번호】	134-060
【주소】	서울특별시 강동구 둔촌동 둔촌주공아파트 114동 404호
【국적】	KR
【발명자】	
【성명의 국문표기】	명성주
【성명의 영문표기】	MYOUNG, Sung Ju

【주민등록번호】 730526-1637611
【우편번호】 135-884
【주소】 서울특별시 강남구 수서동 708번지 삼익아파트 404동 806호
【국적】 KR
【심사청구】 청구
【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 김동진 (인)
【수수료】
 【기본출원료】 20 면 29,000 원
 【가산출원료】 37 면 37,000 원
 【우선권주장료】 0 건 0 원
 【심사청구료】 18 항 685,000 원
 【합계】 751,000 원
【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 플래시 메모리의 맵 블록에 데이터 블록의 상태를 나타내는 소정의 사상 테이블을 만들고, 이를 참조하여 보다 효율적인 메모리 액세스 및 오류블록 복구가 가능하도록 하는 플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법을 개시한 것이다.

본 발명의 플래시 메모리는 데이터 블록을 구성하는 각 블록에 할당되는 물리 주소와 각 블록의 상태 정보로 이루어지는 사상 테이블인 제 1 사상 테이블과, 오류 블록이 제외된 상기 제 1 사상 테이블의 각 블록에 대한 물리 주소와 논리 주소간의 사상정보로 이루어지는 사상테이블인 제 2 사상 테이블과, 상기 제 2 사상 테이블의 갱신동작을 최소화하기 위해 최근의 사상정보를 지정된 값만큼 기록하고 처리하는 사상테이블인 제 3 사상 테이블로 이루어진 맵 블록을 포함하여 구성되는 것으로서, 이를 통해 플래시 메모리를 보다 효율적으로 액세스하고 관리할 수 있으며 플래시 메모리 사용상의 안정성을 극대화 시킬 수 있는 효과를 제공한다.

【대표도】

도 1

【색인어】

플래시 메모리, 맵 블록, 사상 테이블, 액세스 장치,

【명세서】**【발명의 명칭】**

플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법{FLASH MEMORY, ACCESS APPARATUS AND METHOD USING FLASH MEMORY}

【도면의 간단한 설명】

도 1은 본 발명에 따른 플래시 메모리 및 그에 따른 플래시 메모리 액세스 장치를 나타낸 것이다.

도 2는 플래시 메모리 컨트롤러에서 UT를 생성하는 논리합을 도시한 것이다.

도 3의 a 내지 c는 본 발명에 따른 PAT, BIT, TC, UT에 대한 사상 테이블을 일 예로 도시한 것이다.

도 4는 본 발명의 플래시 메모리 액세스 방법에 따른 플래시 메모리의 초기화 과정을 나타낸 동작 흐름도이다.

도 5는 도 4에 도시된 S150 단계에서 PAT를 독출하는 과정을 나타낸 동작 흐름도이다.

도 6은 본 발명의 플래시 메모리 액세스 방법에 따른 읽기 연산 처리과정을 나타낸 동작 흐름도이다.

도 7은 본 발명의 플래시 메모리 액세스 방법에 따른 쓰기 연산 처리과정을 나타낸 동작 흐름도이다.

도 8은 본 발명의 플래시 메모리 액세스 방법에 따른 TC 용량이 부족한 경우의 쓰기 연산 처리과정을 나타낸 동작 흐름도이다.

도 9는 본 발명의 플래시 메모리 액세스 방법에 따른 BIT와 TC 용량이 부족한 경우의 쓰기 연산 처리과정을 나타낸 동작 흐름도이다.

도 10은 본 발명의 플래시 메모리 액세스 방법에 따른 이전의 데이터를 수정 또는 추가하는 경우의 쓰기 연산 처리과정을 나타낸 동작 흐름도이다.

도 11은 본 발명의 플래시 메모리 액세스 방법에 따른 오류복구를 위한 연산 처리과정을 나타낸 동작 흐름도이다.

도 12의 a내지 c는 본 발명의 일 실시 예에 따른 사상정보 갱신 과정에서 발생하는 오류 복구 과정을 사상테이블로 나타낸 것이다.

도 13의 a, b는 본 발명의 일 실시 예에 따른 데이터 기록 과정에서 발생하는 오류 복구 과정을 사상테이블로 나타낸 것이다.

* 도면의 주요부분에 대한 부호의 설명 *

100 : 플래시 메모리

110 : 헤더

131 : PAT

133 : BIT

135 : TC

137 : UT

150 : 데이터 블록

300 : 플래시 메모리 컨트롤러

700 : 램

130 : 맵 블록

132 : PAT spare

134 : BIT spare

136 : TC spare

500 : 프로세서

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <24> 본 발명은 플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법에 관한 것으로서, 특히 플래시 메모리(Flash Memory)의 블록 할당 맵(Block Allocation Map:이하, 맵 블록(Map block)이라 칭함)에 데이터 블록(Data Block)의 물리 주소 및 상태를 나타내는 소정의 사상(mapping) 테이블을 만들고, 이를 참조하여 보다 효율적인 메모리 액세스 및 오류 블록 복구가 가능하도록 하는 플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법에 관한 것이다.
- <25> 일반적으로, 플래시 메모리는 데이터의 기록 및 삭제가 자유로운 램(RAM:Random Access Memory)의 장점과 전원의 공급 없이도 저장된 데이터를 보존하는 롬(ROM:Read Only Memory)의 장점을 동시에 지니고 있어, 최근 디지털 카메라, PDA, MP3 플레이어와 같은 휴대용 전자기기의 저장 매체로 널리 사용되고 있다.
- <26> 이 같은 플래시 메모리는 기존의 RAM이나 비휘발성 저장 장치, 마그네틱 등과 마찬가지로 특정 위치에 저장된 데이터를 임의로 접근할 수 있지만, 데이터를 수정하거나 삭제하는 방법에 있어서는 기존의 저장장치와 다르게 블록(block)을 기본단위로 하여 액세스가 이루어진다.
- <27> 즉, 플래시 메모리에 기록된 데이터를 수정 또는 삭제하고자 하면 해당되는 데이터를 포함하는 데이터 블록을 삭제한 후 재 기록해야하는데, 일 예로, 글자나 알파벳을 하

나씩 수정하는 대신 '삭제'키를 눌러 전체 단어를 다 삭제한 후 다시 수정하는 것과 같다.

<28> 이하, 사용되는 용어 가운데 블록은 플래시 메모리에서 한번의 삭제 연산으로 지울 수 있는 단위를 나타내는 것이고, 페이지(page)는 판독/기록 연산 시에 읽거나 기록할 수 있는 데이터 크기를 의미한다.

<29> 플래시 변환 계층(Flash Translation Layer:이하,FTL이라 칭함)은 플래시 메모리를 블록 디바이스처럼 사용할 수 있게 해주는 소프트웨어로서, 일반적으로 플래시 메모리를 효율적으로 관리하기 위한 방법으로 블록(또는 섹터) 재사상(remapping)기법을 지원한다

<30> 블록(또는 섹터) 재사상 기법은 수정 또는 삭제로 인하여 플래시 메모리에 기록된 데이터의 물리적인 블록번호(Physical Block Number:이하, PBN이라 칭함)가 변경되어도 동일한 논리적인 블록번호(Logical Block Number:이하, LBN이라 칭함)로 플래시 메모리에 기록된 데이터에 접근할 수 있도록, 특정 데이터에 대한 LBN와 PBN간의 사상 정보를 관리하는 것이다.

<31> 그러나 플래시 메모리는 메모리 셀의 수명이 약 100,000 기록 사이클 밖에 안 되기 때문에 부분 기록(Partial Write) 횟수가 제한되어 있다.

<32> 또한, 쓰기 연산 수행과정에서 전원 중단이 발생하는 경우, 기록되던 사용자의 데이터나 재사상을 위해 사용되는 데이터가 손실될 수 있는 문제점이 있었다.

- <33> 따라서 종래에는 블록에 메모리 상태 정보를 기록하고 연산 수행시마다 상태 정보를 적절히 수정하도록 하였는데, 사상 정보를 저장하고 관리하기 위한 저장 공간과 시간이 소요되므로 효율적인 처리 방법이 요구되었다.
- <34> 일 예로, 현재 상용화되고 있는 M-Systems의 NOR 방식 FTL 또는 NAND 방식 FTL은 별도의 대체 페이지 맵(replacement Virtual Page Map: 이하, replacement VPM이라 칭함)을 사용하여 기록(Write) 연산 수행시마다 변경된 사상 정보를 기록해야하기 때문에 모든 데이터 블록마다 replacement VPM을 갖게 되므로 최대 플래시 메모리의 50%가 낭비되는 문제점이 있었다.
- <35> 이러한 종래 기술의 문제점을 보완하기 위하여 앞서 제안된 바 있는, 한국 등록특허 제10-0319598호(발명의 명칭: 플래시 메모리 어레이 액세스 방법 및 장치)는 메모리를 블록 단위로 나누고 각 블록마다 룩업(look up) 테이블을 두어 메모리 액세스 효율을 높이는 액세스 방법 및 장치를 제시한 것이나, 각 블록마다 룩업 테이블을 별도로 구비해야하기 때문에 플래시 메모리의 저장 공간이 효율적으로 사용되지 못하게 되는 문제점이 있었다.
- <36> 또한, 한국 공개특허 제 2001- 0029171호(발명의 명칭: 플래시 메모리와 그 제어방법)는 플래시 메모리에 구비되는 유닛을 통해 데이터를 효율적으로 기록 및 갱신할 수 있도록 하는 플래시 메모리와 제어 방법을 제시한 것이나, 소정의 블록으로 이루어지는 유닛 단위의 데이터 사상을 제시하고 있을 뿐 블록 데이터를 보다 효율적으로 액세스하기 위한 재사상 방법은 제시하지 못하고 있다.
- <37> 그리고, 한국 공개특허 제 2002-0028624호(발명의 명칭: 플래시 메모리를 위한 재사상 제어방법 및 그에 따른 플래시 메모리 구조)는 부분기록을 최소화하면서 블록과 유

니트의 상태를 기록하고 처리할 수 있는 재사상 제어방법 및 플래시 메모리 구조를 제시한 것이나, 호스트에 의한 삭제, 변경 또는 쓰기 등의 연산동작으로 사상관계가 변경될 때마다 맵 블록에 기록된 논리적/물리적 블록 번호간의 사상정보를 검색하여 해당되는 블록의 사상정보를 계속적으로 갱신시켜야하기 때문에 충분한 램 용량이 요구되므로 효율적인 플래시 메모리 관리방법을 제시하지는 못하고 있다.

【발명이 이루고자 하는 기술적 과제】

<38> 본 발명은 상기한 문제점들을 해결하기 위해 안출된 것으로, 플래시 메모리에서 판독 또는 기록 연산 시 플래시 메모리의 맵 블록에 구비되는 소정의 사상 테이블을 참조하여 판독 또는 기록이 요구되는 플래시 메모리의 물리 주소를 효율적으로 액세스할 수 있도록 하는 것을 목적으로 한다.

<39> 본 발명의 다른 목적은 플래시 메모리의 오류 블록 및 갑작스런 전원 중단으로 인해 발생된 오류를 감지하고 데이터 복구를 효과적으로 할 수 있도록 하는

<40> 본 발명의 또 다른 목적은 효율적인 메모리 액세스 및 오류 복구가 가능한 안정적인 플래시 메모리를 제공하는 것이다.

【발명의 구성 및 작용】

<41> 이하, 본 발명에 따른 플래시 메모리, 그에 따른 플래시 메모리 액세스 장치 및 방법을 첨부된 도면을 참조하여 상세히 설명한다.

<42> 본 발명에 따른 플래시 메모리는 데이터 블록을 구성하는 각 블록에 할당되는 물리 주소와 각 블록의 상태 정보로 이루어지는 사상 테이블인 제 1 사상 테이블과, 오류 블록이 제외된 상기 제 1 사상 테이블의 각 블록에 대한 물리 주소와

논리 주소간의 사상정보로 이루어지는 사상테이블인 제 2 사상 테이블과, 제 2 사상 테이블의 갱신동작을 최소화하기 위해 최근의 사상정보를 지정된 값만큼 기록하고 처리하는 사상테이블인 제 3 사상 테이블로 이루어진 맵 블록을 포함하는 것을 특징으로 한다.

<43> 또한, 제 1 사상 테이블, 제 2 사상 테이블, 제 3 사상 테이블에 대한 각각의 여유 블록을 더 포함하는 것을 특징으로 한다.

<44> 본 발명에 따른 플래시 메모리 액세스 장치는 데이터 블록의 물리 주소 정보로 이루어지는 제 1 사상 테이블과 오류블록이 제외된 제 1 사상 테이블의 사상정보로 이루어지는 제 2 사상 테이블과 가장 최근의 사상정보로 이루어지는 제 3 사상 테이블과 각각의 여유 블록으로 이루어지는 맵 블록을 포함하는 플래시 메모리, 플래시 메모리의 맵 블록으로부터 얻어지는 제 1 및 제 2 및 제 3 사상 테이블을 통해 자유블록 정보로 이루어지는 제 4 사상 테이블을 생성하고 판독 및 기록 연산 시 제 2 및 제 3 사상 테이블 또는 제 4 사상 테이블을 참조하여 데이터를 판독/기록할 물리 주소를 액세스하는 플래시 메모리 컨트롤러를 포함하는 것을 특징으로 한다.

<45> 또한, 플래시 메모리 컨트롤러는 제 2 사상 테이블과 제 3 사상 테이블 및 제 4 사상 테이블을 통해 사상정보 갱신 또는 데이터 기록 과정에서의 전원 중단에 의한 오류를 감지하고 데이터를 복구하는 것을 특징으로 한다.

<46> 본 발명에 따른 플래시 메모리 액세스 방법은 프로세서로부터 판독 및 기록 연산이 요구되면 커맨드와 함께 논리 주소를 입력받는 단계; 판독 및 기록 연산을 위해 플래시 메모리로부터 독출되는 데이터 블록의 물리 주소 정보 가운데 오류블록이 제외된 사상정보로 이루어지는 제 2 사상 테이블과 가장 최근의 사상정보로 이루어지는 제 3 사상 테이블에서 상기 논리 주소를 체크하는 단계; 제 2 사상 테이블과 제 3 사상 테이블에 상

기 논리 주소가 존재하면 지정되는 데이터 블록의 물리 주소를 액세스 하여 판독 및 기록 연산을 처리하는 단계를 포함하는 것을 특징으로 한다.

<47> 또한, 제 2 사상 테이블과 제 3 사상 테이블에 상기 논리 주소가 존재하지 않으면 판독 또는 기록 연산에 따라 판독 오류 처리하거나 플래시 메모리 컨트롤러에 의해 생성되는 자유블록 정보로 이루어지는 제 4 사상 테이블을 통해 할당되는 자유블록의 물리 주소를 액세스하여 기록 연산을 처리하는 단계를 더 포함하는 것을 특징으로 한다.

<48> 그리고, 프로세서로부터 요구되는 연산 동작에 따라 상기 플래시 메모리를 효율적으로 액세스하기 위한 사상정보를 플래시 메모리의 맵 블록으로부터 로딩하여 플래시 메모리를 초기화하는 단계를 더 포함하는 것을 특징으로 한다.

<49> 그리고, 기록 연산 과정에서 발생하는 전원 중단에 의한 오류를 감지하고 데이터를 복구하는 단계를 더 포함하는 것을 특징으로 한다.

<50> 이하, 본 발명의 구성 및 동작을 일 실시 예를 통해 설명을 위하여, 제 1 사상 테이블은 물리적 주소 테이블(Physical Address Table: 이하, PAT라 칭함), 제 2 사상 테이블은 블록 정보 테이블(Block Information Table: 이하, BIT라 칭함), 제 3 사상 테이블은 사상 정보 테이블(Table Cache : 이하, TC라 칭함), 제 4 사상 테이블은 미사용 블록 테이블(Unused Table: 이하, UT라 칭함), 각각의 사상 테이블에 대한 여유 블록은 spare(PAT spare, BIT spare, TC spare)로 정의한다.

<51> 우선, 본 발명에 따른 플래시 메모리의 구조 및 그에 따른 플래시 메모리 액세스 장치의 구성과 동작을 첨부된 도면을 참조하여 일 실시 예로 상세히 설명한다.

- <52> 도 1은 본 발명에 따른 플래시 메모리 및 그에 따른 플래시 메모리 액세스 장치를 나타낸 것이다.
- <53> 도 1에 도시된 바와 같이, 본 발명에 따른 플래시 메모리(100)는 헤더(Header)(110), 맵 블록(130), 데이터 블록(150)으로 구성된다.
- <54> 헤더(110)는 플래시 메모리 모델, 메모리 용량, 데이터 블록(150) 관리를 위한 정보 등과 같은 플래시 메모리에 대한 전반적인 정보 및 PAT 주소 정보가 저장된 블록 정보가 저장된다.
- <55> 이러한 헤더(110)에 메인 펌웨어(main firmware)가 구비되면, 플래시 메모리 초기화를 위한 데이터가 보다 빠르게 로딩되고 플래시 메모리 초기화 과정이 효율적으로 이루어진다.
- <56> 맵 블록(130)은 데이터 블록(150)을 통한 데이터 판독/기록/삭제 연산을 수행하기 위한 사상정보가 기록된다.
- <57> 데이터 블록(150)은 데이터 블록(150)은 블록 단위로 액세스될 수 있도록 FTL에 의해 일정한 크기의 블록으로 분할된 다수의 블록으로 이루어지는 것으로, 각 블록은 순차적으로 부여되는 PBN을 갖는다.
- <58> 이와 같이 이루어지는 플래시 메모리(100) 가운데 도 1에 별도로 도시되는 본 발명에 따른 맵 블록(130)의 구조를 첨부된 도면을 참조하여 상세히 설명한다.
- <59> 맵 블록(130)은 도 1에 도시한 바와 같이, PAT(131), BIT(133), TC(135) 및 각 사상 테이블의 갱신 정보를 기록하기 위해 구비되는 여유 블록인 PAT spare(132)/BIT spare(134)/TC spare(136)로 이루어진다.

- <60> PAT(131)는 데이터 블록(150)을 구성하는 다수의 블록, 각 블록에 할당되는 PBN 및 각 블록의 상태 정보로 이루어지는 사상 테이블이다.
- <61> 이러한 PAT(131)는 오류블록이 발생될 때마다 갱신되고, 갱신되는 내용은 현재 PAT가 기록된 페이지의 다음 페이지에 기록되며 이전의 페이지에는 일 예로, 0x00와 같은 코드로 내용이 변경되었음을 표시한다.
- <62> 또한, 플래시 메모리 초기화 과정에서 현재 사용되어야 할 PAT(131)를 확인할 수 있도록 내용이 변경될 때마다 변경되는 PAT 주소 정보를 헤더(110)에 기록한다.
- <63> BIT(133)는 오류 블록이 제외된 PAT(131)의 각 블록에 대한 LBN와 PBN간의 사상정보로 이루어지는 사상테이블이다.
- <64> 이러한 BIT(133)는 효율적인 기록 연산을 위한 것으로, 프로세서(500)로부터 요구되는 기록 연산 시 플래시 메모리 컨트롤러(300)에서 BIT(133)를 통해 쓰기 가능한 비어있는 블록(이하, 자유블록(free block)이라 칭함)을 확인하고 데이터를 기록함으로써 데이터 블록에서 일일이 자유블록을 검색하지 않고 빠르게 기록 연산을 수행할 수 있도록 한다.
- <65> 또한, PAT(131)와 같이 새로운 사상정보가 갱신될 때마다 BIT 내용을 변경하고, 변경되는 BIT를 현재 BIT가 기록된 페이지 다음 페이지에 기록하며 BIT 사용영역을 나타낸 포인터를 변경시킨다.
- <66> TC(135)는 BIT(133)의 갱신동작을 최소화하기 위해 최근의 사상정보를 지정된 값만큼 기록하고 처리하는 사상테이블이다.

- <67> 이러한 TC(135)는 기록/삭제 연산 동작마다 매번 BIT(313)를 갱신시킴으로써 발생되는 부하를 최소화하기 위한 것으로, 프로세서(500)로부터 요구되는 기록/삭제 연산에 의해 발생하는 최근의 사상정보를 지정된 값만큼 기록하고 사상정보가 지정된 값에 도달하면 BIT(133)로 갱신함으로써 BIT(133)를 기록/삭제시마다 매번 갱신하지 않고 갱신 동작을 최소화하게 된다.
- <68> 또한, BIT(133)의 반복적인 갱신동작으로 인한 기록 속도 저하도 막을 수 있다.
- <69> 일 예로, TC(135)에 기록되는 사상정보가 3개로 지정된 경우, 프로세서(500)로부터 요구되는 기록/삭제 연산에 따라 TC(135)에 기록되는 사상정보가 3개에 이르면 TC(315)에 기록된 3개의 사상정보는 BIT(313)에 옮겨 기록된다.
- <70> PAT spare(132)/BIT spare(134)/TC spare(136)는 PAT(131)/BIT(133)/TC(135)의 특정 블록이 기록/삭제 연산에 의해 반복적으로 사용되는 것을 막기 위해 구비된다.
- <71> 이 가운데 BIT spare(134)/TC spare(136)는 기록/삭제 연산에 따라 상기 BIT(133) 및 TC(135)의 사상 정보를 갱신하는 과정에서 발생하는 갑작스런 전원 중단에 의한 오류 발생에 대비할 수 있도록 갱신되는 기존의 BIT 및 TC의 사상정보를 저장한다.
- <72> 일 예로, 프로세서(500)로부터 요구되는 기록연산 처리과정에서 발생하는 새로운 사상정보를 TC(135)에 기록하고자 하는 경우, TC(135)에 기록된 이전의 사상정보가 할당된 저장 용량에 도달하거나 지정된 값에 도달하면 TC(135)의 사상정보를 BIT(133)로 옮겨 기록한다.
- <73> 이 경우, 기존의 사상정보를 BIT(133)로 갱신하기 전에 TC spare(136)에 기존의 사상정보를 기록함으로써, 플래시 메모리 컨트롤러(300)에서 TC(135)의 사상정보를

BIT(133)에 기록하는 과정에서 발생하는 갑작스런 전원 중단으로 인한 오류를 확인하고 오류 블록을 처리할 수 있도록 한다.

<74> 또 다른 예로, 사상정보 BIT(133)에 기록된 이전의 사상정보가 할당된 저장 용량에 도달하면 BIT(133)블록에서 다음 페이지로 BIT 사상정보를 옮겨 기록한다.

<75> 이 경우, 기존의 사상정보를 다음 페이지로 갱신하기 전에 BIT spare(134)에 기존의 사상정보를 기록함으로써, 플래시 메모리 컨트롤러(300)에서 BIT(133)의 사상정보를 다음 페이지에 기록하는 과정에서 발생하는 갑작스러운 전원 중단으로 인한 오류를 확인하고 오류 블록을 처리할 수 있도록 한다.

<76> 이와 같이 구성되는 플래시 메모리(100)를 이용하여 프로세서(500)로부터 요구되는 판독/기록 연산 시마다 플래시 메모리(100)를 효율적으로 액세스하기 위한 플래시 메모리 액세스 장치는 도 1에 도시한 바와 같이, 크게 플래시 메모리(100)와, 플래시 메모리 컨트롤러(300)로 구성되며, 본 발명을 일 실시 예에 따른 구현을 위해 프로세서(500), 램(700)을 더 포함한다.

<77> 본 발명에 따른 플래시 메모리 컨트롤러(300)는 플래시 메모리(100)의 맵 블록(130)으로부터 얻어지는 데이터 블록(150)에 대한 물리 주소 및 사상 정보로 이루어지는 PAT(131)/BIT(133)/TC(135)를 통해 미사용 블록 테이블(Unused Table : 이하, UT라 칭함)(137)을 생성하고 판독/기록 연산 시 BIT(133)/TC(135) 또는 UT(137)를 참조하여 데이터를 판독/기록할 물리 주소를 액세스한다.

<78> 프로세서(500)는 플래시 메모리(100)에 기록된 데이터를 판독하거나 데이터를 기록하기 위한 제어 명령어를 발생시켜 플래시 메모리 컨트롤러(300)로 전송한다. 램(700)은

프로세서(500) 및 플래시 메모리 컨트롤러(300)에 의한 플래시 메모리(100) 액세스 과정에서 얻어지는 데이터를 저장한다.

<79> 플래시 메모리 컨트롤러(300)에 의해 생성되는 UT(137)는 도 2에 도시된 바와 같이, PAT(131), BIT(133) 및 TC(135)를 논리합으로 병합하는 것으로, BIT(133)와 TC(137)에 기록된 사상정보가 추가된 PAT(131)로 이루어진다.

<80> 이러한 과정을 통해 생성된 UT(137)는 플래시 메모리 컨트롤러(300)에 의해 상대적으로 빠른 읽기가 가능한 램(700)에 저장되어 기록 연산 시 자유블록 검색이 빠르도록 하는데, 이 때문에 플래시 메모리(100)에 기록되는 PAT(131), BIT(133), TC(135)와는 달리 플래시 메모리(100)를 초기화 시킬 때마다 플래시 메모리 컨트롤러(300)에 의해 생성된다.

<81> 이와 같이 이루어지는 본 발명에 따른 플래시 메모리(100) 및 플래시 메모리 액세스 장치에서 PAT(131), BIT(133), TC(135), UT(137)를 첨부된 도면을 참조하여 일 실시예로 상세히 설명한다.

<82> 도 3의 a 내지 c는 본 발명에 따른 PAT(131), BIT(133), TC(135), UT(137)에 대한 사상 테이블을 일 예로 도시한 것이다.

<83> PAT(131)는 도 3의 a에 도시된 바와 같이, 데이터 블록(150)의 물리 주소 정보를 나타내는 것으로, FTL에 의해 블록 디바이스와 같이 일정한 크기의 블록으로 분할된 다수의 블록, 각 블록에 순차적으로 할당되는 PBN, 각 블록에 기록되는 상태 정보로 이루어진다.

- <84> 각 블록의 상태 정보는 정상적인 블록(Good Block)을 나타내는 '0'과 사용하지 못하는 오류 블록(Bad Block)을 나타내는 '1'로 이루어진다.
- <85> BIT(133)는 도 3의 b에 도시된 바와 같이, 데이터 블록(150)의 블록들에 대한 사용 정보를 나타내는 것으로, 오류 블록이 제외된 PAT(131)의 각 블록에 대한 LBN와 PBN간의 사상정보로 이루어진다.
- <86> 즉, 도시된 바와 같이, 프로세서(500)로부터 기록이 요구되는 LBN에 사상되는 특정 PBN, 사용되지 않는 LBN을 나타내는 소정의 코드(일 예로, 0xFFFF)가 기록된다.
- <87> TC(135)는 도 3의 b에 도시된 바와 같이, 프로세서(500)로부터 요구되는 기록연산 처리결과 발생되는 최근의 사상정보로 이루어진다.
- <88> 이러한 TC(135)의 사상정보는 기록되는 사상정보가 제한되어 지정된 값만큼 기록되면, BIT(133)로 갱신된다.
- <89> 일 예로, TC(135)에 기록되는 사상정보가 3개로 지정된 경우, 프로세서(500)에 의한 기록/삭제 연산에 따라 TC(135)에 기록되는 사상정보가 3개에 이르면 TC(315)에 기록된 3개의 사상정보는 BIT(313)에 옮겨 기록된다.
- <90> 이에 따라, BIT(313)는 TC(315)로부터 사상정보가 옮겨질 때에만 다음 페이지로 갱신되므로, 기록/삭제 연산 시마다 매번 갱신하지 않고 갱신 동작을 최소화하게 된다.
- <91> 또한, BIT(133)와 TC(135)는 단계적 사상 기법을 통해 LBN과 PBN을 1:1로 사상하지 않고, 프로세서(500)로부터 데이터 기록이 요구되면 임의의 자유블록을 데이터를 기록할 블록으로 할당하여 입력되는 데이터를 기록한 후 프로세서(500)에 의해 요구된 논리 블록 주소가 사상된 물리 블록 주소를 저장한다.

- <92> 도 3의 b를 참조하여 이러한 BIT(133)와 TC(135)간의 사상정보 갱신 과정을 일 예로 상세히 설명한다.
- <93> 만일, 플래시 메모리가 초기상태이면(갱신 전) BIT(133)와 TC(135)는 사용되지 않는 상태(0xFFFF)인 LBN으로 이루어지고, 이러한 초기 상태에서 프로세서(500)로부터 기록 연산이 요구될 때마다 새로운 LBN과 PBN간의 사상 정보는 TC(135)에 순차적으로 기록된다.
- <94> 예를 들어, TC(135)의 첫 번째 LBN(0)에는 프로세서(500)로부터 입력되는 데이터가 기록된 PBN인 '1'이 기록되는데, '1'은 프로세서(500)에 의해 기록이 요구되는 논리 블록 주소가 사상되는 물리 블록 정보에 해당된다.
- <95> 이와 같이, 두 번째 LBN(1)에는 프로세서(500)로부터 입력되는 데이터가 기록된 PBN인 '4'가 기록되고, 세 번째 LBN(2)에는 프로세서(500)로부터 입력되는 데이터가 기록된 PBN인 '56'이 기록된다.
- <96> 만일, TC(315)에 저장 가능한 사상정보가 3개로 지정되어 있다면, TC(315)의 사상정보가 지정된 값에 도달했으므로 TC(315)의 사상정보는 BIT(313)로 갱신된다.
- <97> 이에 따라, BIT(313)에는 TC(135)로부터 갱신되는 사상정보가 해당되는 각각의 LBN에 기록되는데, 각 LBN에 해당되는 PBN가 기록되고, TC(315)의 사상정보는 삭제된다(갱신 후).
- <98> UT(137)는 도 3의 C에 도시된 바와 같이, 데이터 블록(150)의 블록들 가운데 사용 가능한 블록 정보를 나타내는 것으로, BIT(133) 및 TC(135)의 사상정보가 추가되는 PAT(131)로 이루어진다.

- <99> 이러한 UT(137)는 프로세서(500)로부터 요구되는 기록 연산에 따라 데이터를 기록할 자유블록을 제공하는 것으로, 보다 효율적인 자유블록을 액세스할 수 있도록 포인트로 첫 자유블록을 지시한다.
- <100> 이와 같은 본 발명에 따른 플래시 메모리와 플래시 메모리 액세스 장치를 이용하여 플래시 메모리를 액세스하는 방법을 첨부된 도면을 참조하여 일 실시 예로 상세히 설명한다.
- <101> 먼저, 플래시 메모리 컨트롤러(300)를 통해 플래시 메모리(100)를 초기화하는 과정을 설명하고, 초기화된 플래시 메모리(100)에서 프로세서(500)로부터 요구되는 판독/기록 연산을 수행하는 처리과정을 설명한다.
- <102> 1. 초기화 과정
- <103> 초기화 과정은 프로세서(500)로부터 요구되는 연산 동작에 따라 플래시 메모리(100)를 효율적으로 액세스하기 위한 사상정보를 플래시 메모리(100)로부터 로딩하는 것이다.
- <104> 도 4는 본 발명에 따른 플래시 메모리 초기화 과정을 나타낸 동작 흐름도이고, 도 5는 도 4의 PAT(131) 독출 과정을 상세히 나타낸 것이다.
- <105> 도 4에 도시된 바와 같이, 우선, 플래시 메모리 컨트롤러(300)에서는 플래시 메모리(100)의 헤더(110)에 기록된 플래시 메모리에 대한 전체 정보 및 PAT 주소정보가 기록된 블록의 주소정보와 같은 플래시 정보를 체크하고(S110), 플래시 메모리(100)의 맵 블록(130)에 기록된 PAT(131), BIT(133) 및 TC(135)를 독출하기 위한 변수를 초기화(Init Variables) 한다(S130).

- <106> 그 다음, 헤더(110)의 특정 블록으로부터 얻어지는 PAT 주소정보를 이용하여, 플래시 메모리(100)의 맵 블록(130)에 할당된 PAT 블록 영역 가운데 현재 사용되어야 할 PAT가 저장된 블록을 검색하고, 해당 블록에 기록된 PAT(131)를 독출한다.
- <107> 또한, 플래시 메모리(100)의 맵 블록(130)에 할당된 BIT/TC 블록 영역의 모든 블록을 검색하여 BIT(133) 및 TC(135)를 독출한다(S150).
- <108> 이와 같이, PAT(131), BIT(133), TC(135)가 독출되면, 플래시 메모리 컨트롤러(300)에서는 독출된 PAT(131), BIT(133) 및 TC(135)를 논리합으로 병합시켜 UT(137)를 생성하고, 생성된 UT(137)를 램(700)에 저장한다(S170).
- <109> 이러한 초기화 과정이 완료되면, 프로세서(500)로부터 요구되는 판독 또는 기록 연산에 따른 연산동작을 수행하기 위해 대기한다(S190).
- <110> 이러한 초기화 과정 가운데 PAT(131)를 독출하는 단계(S130)를 첨부된 도 5를 참조하여 상세히 설명한다.
- <111> 도 5에 도시된 바와 같이, 우선, 플래시 정보를 체크하는 단계(S110)에서 헤더(110)로부터 얻어지는 PAT 주소정보가 기록된 블록에 대한 주소정보를 이용하여 헤더(110)의 블록들 가운데 지정된 블록에서 PAT 주소정보를 검색한다(S151).
- <112> PAT 주소정보 검색 결과를 통해 PAT 주소정보가 존재하는지 여부를 판별하여(S152), 판별결과 PAT 주소정보가 존재하면 검색된 PAT 주소정보를 판독하고(S153), 판별결과 PAT 주소정보가 존재하지 않으면 맵 영역(130)에 할당된 PAT 블록영역의 모든 블록에서 PAT를 검색한다(S154).

<113> PAT 블록영역의 모든 블록에 대한 PAT 검색 결과를 통해 PAT가 PAT 블록 영역에 존재하는지 여부를 판별하여(S155), 판별결과 PAT(131)가 검색되면 PAT(131)가 검색된 블록의 주소정보를 헤더(110)의 지정된 블록에 PAT 주소정보로 기록한다(S156).

<114> 또한, 판별결과 PAT(131)가 존재하지 않으면 데이터 블록(150)의 각 블록들에 PBN을 할당하고 각 PBN의 상태를 검사하여 새로운 PAT(131)를 생성하고, 생성된 PAT(131)를 PAT 블록 영역의 특정 블록에 기록한 후 헤더(110)의 지정된 블록에 PAT(131)가 기록된 블록 주소를 PAT 주소정보로 기록한다(S157).

<115> 이러한 과정을 통해 PAT 주소정보가 생성되거나 독출되면 플래시 메모리 컨트롤러(300)에서는 독출된 PAT 주소정보를 이용하여 PAT 블록 영역에서 PAT(131)를 독출한다(S158).

<116> 2. 판독 연산 과정

<117> 판독 연산 과정은 프로세서(500)로부터 소정의 데이터에 대한 판독이 요구될 때에 판독 커맨드와 함께 논리 주소가 입력되면, 플래시 메모리(100)의 데이터 블록(150)에서 해당되는 물리 주소를 검색하여 기록된 데이터를 프로세서(500)로 제공하는 것이다.

<118> 도 6은 본 발명의 플래시 메모리 액세스 방법에 따른 판독 연산 처리과정을 나타낸 동작 흐름도이다.

<119> 도 6에 도시된 바와 같이, 우선, 플래시 메모리 컨트롤러(300)에서는 프로세서(500)로부터 판독 동작 요구와 함께 해당되는 데이터의 논리 주소인 소정의 논리 블록 주소가 입력되면(S300), BIT(133)와 TC(135)를 참조하여 입력된 논리 블록 주소를 플래시 메모리(100)의 물리 블록 주소로 변환한다(S301)

- <120> 이와 같이 논리 주소가 물리 주소로 변환되면, 변환된 물리 주소를 통해 프로세서(500)로부터 판독 요구되는 논리 주소가 실제 존재하는 주소인지 여부를 확인하기 변환된 물리 주소가 데이터 블록(150)의 물리 주소범위에 존재하는지 여부를 판별하여(S302), 판별결과 존재하지 않는 물리 주소이면 프로세서(500)로 판독 오류 코드를 포함하는 메시지를 전송하여 오류 처리한다(S303).
- <121> 판별결과 존재하는 물리 주소이면 변환된 물리 주소를 이용하여 플래시 메모리(100)의 데이터 블록(150)에서 해당되는 물리 주소를 검색하고(S304), 검색된 물리 주소에 기록된 데이터를 독출하여 프로세서(500)로 전송한다(S305).
- <122> 3. 기록 연산 과정
- <123> 기록 연산 과정은 프로세서(500)로부터 소정의 데이터에 대한 기록이 요구될 때에 기록 커맨드와 함께 논리 주소가 입력되면, 첫 자유블록 또는 단계적 사상기법에 의해 할당되는 물리 주소를 데이터가 기록될 블록으로 결정하고 논리 주소와 데이터가 기록되는 물리 주소를 기록 연산에 따른 사상정보로 TC(135)에 기록하는 것이다.
- <124> 이러한 기록 연산은 맵 블록(130) 또는 데이터 블록(150)의 상태에 따라 크게, 고려해야 될 사항이 거의 없는 경우, 맵 블록(130)에 할당된 TC 블록 영역에 기록된 사상정보로 인해 TC(135) 저장용량이 부족한 경우, TC(135) 및 BIT(133) 저장용량이 부족한 경우, 기록 요구되는 논리 주소에 존재하는 기존에 기록된 데이터를 수정 또는 추가하는 경우로 이루어진다.
- <125> 먼저, 고려해야 될 사항이 거의 없는 순수한 기록 연산 처리과정을 설명하고, 순차적으로 각 상황에 따른 기록 연산 처리 과정을 설명한다.

- <126> 도 7은 본 발명에 따른 플래시 메모리 액세스 방법에 따른 순수한 쓰기 연산 처리 과정만을 나타낸 동작 흐름도이다.
- <127> 도 7에 도시된 바와 같이, 플래시 메모리 컨트롤러(300)에서는 프로세서(500)로부터 기록 연산이 요구되고 소정의 논리 블록 주소가 입력되면(S510), 입력된 논리 블록 주소를 통해 해당되는 블록 주소가 BIT(133)/TC(135)에 존재하는지 여부를 판별한다(S511).
- <128> 판별결과 논리 블록 주소가 BIT(133)/TC(135)에 존재하지 않으면 UT(137)를 참조하여 포인터로 지시되는 쓰기 가능한 첫 자유 블록을 검색하고(S512), 판별결과 논리 블록 주소가 BIT(133)/TC(135)에 존재하면 단계적 사상기법을 통해 할당되는 임의의 블록을 데이터를 기록할 블록으로 선택한다.
- <129> 이러한 과정을 통해 데이터를 기록할 블록으로 선택되는 첫 자유 블록 또는 단계적 사상에 의해 할당되는 임의의 블록에 해당되는 물리 주소와 프로세서(500)에 의해 쓰기 요구되는 논리 주소간의 사상정보를 TC(135)에 기록하고 새로 기록되는 사상정보에 따라 변경되는 사용 영역을 나타내는 TC(135) 포인터를 변경한다(S513).
- <130> 일 예로, 프로세서(500)로부터 기록이 요구되는 논리 주소가 BIT(131)/TC(135)에 존재하지 않는 경우, 플래시 메모리 프로세서(300)에서는 UT(137)의 첫 자유블록을 데이터를 기록할 블록으로 사용하기 위해 다음과 같이 정의된 함수를 호출한다.
- <131> Get_Free_Block()
- <132> return value is this block number
- <133> This Block is the first unused block at UT

- <134> 이와 같은 함수가 호출됨에 따라, 도 3의 c에 도시된 UT(137)에서 포인터로 지시되고 있는 첫 자유블록이 검색된다.
- <135> 이와 같이, 데이터를 기록한 블록이 결정되고 TC(135)의 사상정보가 변경되면, 버퍼를 통해 프로세서(500)로부터 데이터를 입력받는다(S514).
- <136> 플래시 메모리 프로세서(300)에서는 데이터 입력이 완료되면 입력받은 데이터를 플래시 메모리(100)의 데이터 블록(150) 가운데 데이터를 기록할 블록으로 결정된 물리 주소에 기록한다(S515).
- <137> 이러한 과정을 통해 데이터를 기록하면서 플래시 메모리 컨트롤러(300)에서는 갑작스런 전원 중단에 의한 오류가 발생하는지 여부를 판별한다(S516).
- <138> 판별결과에 따라 데이터 기록을 완료하거나, 데이터를 기록하던 블록을 오류 블록으로 지정한 후 PAT(131)에 오류 블록을 기록하고 변경된 PAT(131)를 현재 PAT가 기록된 다음 페이지에 기록한다.
- <139> 또한, 오류 블록에 의해 변경된 PAT 주소정보를 헤더(110)에 기록하고, 데이터 기록을 완료하기 위해 UT(137)를 참조하여 데이터를 기록할 첫 자유블록을 검색하는 루틴으로 리턴한다(S517).
- <140> 도 8은 TC(135)의 용량이 부족한 경우에 이루어지는 본 발명의 플래시 메모리 액세스 방법에 따른 쓰기 연산 처리 과정을 나타낸 동작 흐름도이다.
- <141> TC(135) 용량이 부족한 쓰기 연산 과정은 도 9의 일반적인 쓰기 연산과정과 유사하므로 중복되는 설명은 생략한다.

- <142> 도 8에 도시한 바와 같이, 우선, 프로세서(500)로부터 기록 연산이 요구되면 해당되는 커맨드와 함께 소정의 논리 블록 주소를 입력받는다(S530).
- <143> 그 다음, 플래시 메모리 컨트롤러(300)에서는 입력된 논리 주소가 BIT(133)/TC(135)에 존재하는지 여부를 판별하고(S531), 판별 결과에 따라 UT(137)를 참조하여 사용 가능한 첫 자유 블록을 검색하거나(S532), 단계적 사상기법에 의해 할당되는 임의의 블록을 통해 데이터를 기록할 블록을 검색한다(S533).
- <144> 이러한 과정에 따라 첫 자유블록 또는 임의의 블록이 데이터를 기록할 블록으로 결정되면 이에 따른 사상정보를 TC(135)에 기록한다.
- <145> 그러나, TC(135)의 저장용량이 부족하므로 맵 블록(130)의 BIT 블록 영역 가운데 새로운 블록을 할당받아 TC(135)로부터 갱신되는 사상정보로 변경되는 BIT(133)를 기록하고(S534), 이전의 TC(135)를 삭제한다(S535).
- <146> 이와 같이 새로 할당된 블록에 갱신되는 BIT(133)를 기록하고 이전의 TC 블록을 삭제함에 따라, BIT(133) 및 TC(135)의 블록이 변경되어 BIT 또는 TC 블록 영역 내에서 사용되고 있는 영역을 지시하는 BIT(133)와 TC(135) 포인터를 변경한다(S536).
- <147> BIT(133)와 TC(135)의 포인터를 변경한 후 플래시 메모리 컨트롤러(300)에서는 기록 연산에 따라 프로세서(500)로부터 전송되는 데이터를 버퍼로 입력받고(S537), 지정된 플래시 메모리(100)의 물리 주소에 기록한다(S538).
- <148> 또한, 데이터 기록과정에서의 오류 발생 여부를 판별하여(S539), 오류가 발생되면 데이터를 기록하던 블록을 오류 블록으로 지정한 후 PAT(131)에 오류 블록을 기록하고 변경된 PAT(131)를 현재 PAT가 기록된 다음 페이지에 기록한다.

- <149> 또한, 오류 블록에 의해 변경된 PAT 주소정보를 헤더(110)에 기록하고, 데이터 기록을 완료하기 위해 UT(137)를 참조하여 데이터를 기록할 첫 자유블록을 검색하는 루틴으로 리턴한다(S540).
- <150> 도 9는 BIT(313) 및 TC(135)의 용량이 부족한 경우에 이루어지는 본 발명의 플래시 메모리 액세스 방법에 따른 쓰기 연산 처리 과정을 나타낸 동작 흐름도이다.
- <151> BIT(313) 및 TC(135) 용량이 부족한 쓰기 연산 과정은 도 8의 TC(135) 용량이 부족한 경우에 이루어지는 쓰기 연산과정과 유사하므로 중복되는 설명은 생략한다.
- <152> 도 9에 도시한 바와 같이, 우선, 프로세서(500)로부터 기록 연산이 요구되면 해당되는 커맨드와 함께 소정의 논리 블록 주소를 입력받는다(550).
- <153> 그 다음, 플래시 메모리 컨트롤러(300)에서는 입력된 논리 주소가 BIT(133)/TC(135)에 존재하는지 여부를 판별하고(S551), 판별 결과에 따라 UT(137)를 참조하여 사용 가능한 첫 자유 블록을 검색하거나(S552), 단계적 사상기법에 의해 할당되는 임의의 블록을 통해 데이터를 기록할 블록을 검색한다(S553).
- <154> 이러한 과정에 따라 첫 자유블록 또는 임의의 블록이 데이터를 기록할 블록으로 결정되면 이에 따른 사상정보를 TC(135)에 기록한다.
- <155> 그러나, BIT(133) 및 TC(135)의 용량이 부족하므로, BIT 블록 영역에서 새로운 블록을 할당받아 TC(135)의 사상정보가 갱신되는 BIT(133)를 기록하고(S554), 이전의 BIT(133) 및 TC(135)를 삭제한다.(S555).
- <156> 이와 같이, 새로운 블록을 할당받아 갱신되는 BIT(133)를 기록하고 이전의 BIT(133)와 TC(135)를 삭제함으로써 BIT(133) 및 TC(135)의 블록이 변경되어 BIT 또는

TC 블록 영역 내에서 사용되고 있는 영역을 지시하는 BIT(133)와 TC(135) 포인터를 변경한다(S556).

<157> 이러한, BIT(133)와 TC(135)의 사상정보 변경 후 프로세서(500)로부터 버퍼로 데이터를 입력받고(S557), 선택된 물리 주소에 해당되는 플래시 메모리(100)의 데이터 블록에 기록한다(S558).

<158> 또한, 데이터 기록 과정에서의 오류 발생 여부를 판별하여(S559), 오류 발생 시 데이터가 기록되던 블록을 오류블록으로 지정한 후 PAT(131)에 기록하고, 변경된 PAT(131)를 현재 PAT가 기록된 다음 페이지에 기록하며 이에 따라 변경되는 PAT 주소정보를 헤더(110)에 기록한다.

<159> 그리고 데이터 기록을 완료하기 위해 UT(137)를 참조하여 데이터를 기록할 첫 자유블록을 검색하는 루틴으로 리턴한다(S560).

<160> 도 10은 기존에 기록된 데이터를 수정 또는 추가하는 경우에 이루어지는 본 발명의 플래시 메모리 액세스 방법에 따른 쓰기 연산 처리 과정을 나타낸 동작 흐름도이다.

<161> 데이터를 수정 또는 추가하는 쓰기 연산 과정은 도 7의 쓰기 연산과정과 유사하므로 중복되는 설명은 생략한다.

<162> 도 10에 도시한 바와 같이, 먼저 플래시 메모리 컨트롤러(300)에서는 데이터를 기록할 블록을 결정하고 이에 따라 변경되는 사상 정보를 TC(135)에 기록 한다.

<163> 그 다음, 프로세서(500)로부터 전송되는 데이터를 버퍼로 입력받고(S575), 입력된 데이터와 기존에 기록된 데이터를 병합하여 플래시 메모리(100)의 데이터 블록(150) 가운데 지정된 물리 주소에 기록한다(S576).

- <164> 이와 같이 수정 또는 추가되는 데이터를 기록한 후에 플래시 메모리 컨트롤러(300)에서는 이전에 데이터가 기록되었던 블록에 기록된 데이터를 삭제한다(S577).
- <165> 또한, 데이터 기록 과정에서 오류 발생 여부를 판별하여 오류 발생 여부에 따라 오류 복구를 한다.
- <166> 4. 오류복구 연산 과정
- <167> 오류복구 연산 과정은 데이터 기록 과정에서 갑작스런 전원 중단과 같은 오류가 발생한 경우, 오류 발생을 감지하고 데이터 복구를 수행하는 것이다.
- <168> 이러한 오류복구 연산은 프로세서(500)로부터 요구되는 기록 연산을 수행하는 과정에서 BIT(133) 또는 TC(135)의 사상정보를 갱신하는 과정에서 전원 중단이 발생하는 경우, 프로세서(500)로부터 입력되는 데이터를 해당 물리 주소에 기록하는 과정에서 발생하는 경우로 이루어진다.
- <169> 본 발명에 따른 오류복구 연산에서는 오류복구를 위해, 먼저 사상정보 갱신하는 과정에서 오류가 발생여부를 판별하여 사상정보를 복구한 후, 데이터를 기록하는 과정에서도 오류가 발생되었는지를 판별하여 데이터를 복구한다.
- <170> 이를 통해 갑작스런 전원 중단이 발생되었을 경우, 사상정보를 기록하는 과정에서의 오류 및 데이터를 기록하는 과정에서의 오류를 모두 판별하고 복구할 수 있게 된다.
- <171> 도 11은 본 발명의 플래시 메모리 액세스 방법에 따른 오류복구를 위한 연산 처리 과정을 나타낸 동작 흐름도이다.

- <172> 도 11에 도시한 바와 같이, 먼저 맵 블록(130)으로부터 얻어지는 BIT(313)/TC(315)를 체크하여(S700), BIT spare(134)/TC spare(136)를 포함하는 모든 BIT/TC 블록 영역에 BIT(133)/TC(135)가 복수개 존재하는지 여부를 판별한다(S701).
- <173> 판별결과 BIT(133)/TC(135)가 복수개 존재하면 사상정보를 갱신하는 과정에서 전원 중단에 의해 오류가 발생되어 BIT spare(134)/TC spare(136)에 사상정보가 기록되어 있는 상태로 판단한다.
- <174> 이러한 경우, 갑작스런 전원 중단에 의한 사상정보 오류를 복구하기 위해 BIT(133)/TC(135)에 기록된 사상정보 가운데 가장 최근의 사상정보를 삭제하고, 최근의 사상정보가 삭제된 BIT(133)/TC(135)와 맵 블록(130)으로부터 얻어지는 PAT(131)를 초기화 과정에서 플래시 메모리(300)에 의해 생성된 UT(137)에 병합시킨다(S702).
- <175> 판별결과 BIT(133)/TC(135)가 하나만 존재하면, 검색된 BIT(133)/TC(135) 및 맵 블록(130)으로부터 얻어지는 PAT(131)를 초기화 과정에서 플래시 메모리 컨트롤러(300)에 의해 생성된 UT(137)에 병합시킨다(S703).
- <176> 이와 같은 과정을 통해 사상정보 갱신과정에서의 오류 발생 여부를 확인하고 사상정보 오류를 복구한 후, UT(137)를 통해 UT(137)의 첫 자유블록에 데이터가 기록되어 있는지 여부를 검색한다(S704).
- <177> UT(137)의 첫 자유블록에 대한 검색 결과를 통해 첫 자유 블록이 순수한 자유블록인지 여부를 판별하여(S705), 첫 자유블록이 순수한 자유블록이 아닌 경우에는 첫 자유블록에 기록된 데이터를 삭제한다(S706).

- <178> 이러한 과정을 통해 프로세서(500)로부터 요구되는 기록연산 수행과정에서 발생되는 오류를 감지하고 데이터를 복구할 수 있다.
- <179> 또한, 이러한 오류복구 연산 과정은 갑작스런 전원 중단 후 인가되는 전원에 의해 플래시 메모리를 초기화 하는 과정뿐만 아니라, 플래시 메모리 컨트롤러(300)의 제어에 따라 수시로 수행될 수 있다.
- <180> 도 12의 a내지 c는 본 발명에 따른 오류 복구 과정 가운데 사상정보 갱신 과정에서의 오류 복구 과정을 나타낸 것으로, 도 12의 a는 BIT(133) 및 TC(135)의 사상 정보가 갱신되기 이전의 초기 상태이고, 도 12의 b는 새로운 사상 정보로 인해 TC(135)의 사상 정보를 BIT(133)로 갱신하는 과정에서의 전원 중단 발생된 상태이고, 도 12의 c는 전원이 인가되어 초기화 과정이 이루어진 상태를 나타낸 것이다.
- <181> 도 12의 a에 도시된 바와 같이, 프로세서(500)로부터 요구되는 기록 또는 삭제 연산에 의해 변경되는 사상정보가 TC(135)에 기록되고, 이러한 과정에서 프로세서(500)로부터 요구된 기록 연산에 의해 새로 생성된 사상정보에 의해 LBN(2)의 사상정보가 갱신되면, 도 12의 b에 도시한 바와 같이 TC(135)의 LBN(2)에 새로운 사상정보가 기록된다.
- <182> 만일, TC(135)에 지정된 저장 가능한 값이 3개의 사상정보라면, TC(135)에 기록된 사상정보가 지정된 값에 도달되었기 때문에 BIT(133)로 사상정보를 갱신한다.
- <183> TC(135)의 사상정보를 BIT(133)로 갱신하기 위해서는 도 12의 b에 도시한 바와 같이, 먼저, 여유블록인 TC spare(136)에 기존의 사상정보를 기록하고, BIT(133)로 사상정보를 갱신한다.

- <184> 이러한 사상정보 갱신 과정에 따라 BIT(133)의 LBN(0)에 데이터를 기록한 후 갑작스런 전원 중단이 발생되었다면, 현재의 상태에서 데이터 기록이 중단된다.
- <185> 이 후, 전원이 인가되면 플래시 메모리 컨트롤러(300)에 의해 플래시 메모리 초기화 과정이 수행되고, PAT(131), BIT(133), TC(135)가 검출된다.
- <186> 플래시 메모리 컨트롤러(300)에서는 초기화 과정을 수행되면 갑작스런 전원 중단에 의한 오류를 감지하고 복구하기 위해, 맵 블록(130)에 할당된 BIT 블록영역 및 TC 블록영역을 검색한다.
- <187> 검색결과 도 12의 c에 도시한 바와 같이, BIT(133), TC(135), TC spare(136)가 검색되었다면, 플래시 메모리 컨트롤러(300)에서는 TC(135)에 해당되는 사상정보가 복수개 검색되었기 때문에 전원 중단으로 인해 오류가 발생된 것으로 판단하고 사상정보 오류를 복구하기 위해 BIT(133) 및 TC(135)에 기록된 최근의 사상정보를 삭제한다.
- <188> 이에 따라, BIT(133)의 LBN(0)에 기록된 PBN는 삭제되고, TC(135)의 LBN(2)에 기록된 PBN도 삭제된다.
- <189> 이와 같은 과정을 통해서, 프로세서(500)로부터 요구되는 기록 연산에 따라 발생된 사상정보를 갱신하는 과정에서의 데이터 오류를 복구한다.
- <190> 도 13의 a 및 b는 이러한 본 발명에 따른 오류 복구 과정 가운데 데이터 기록 과정에서의 오류 복구 과정을 나타낸 것으로, 도 13의 a는 데이터 기록과정에서의 전원 중단을 나타낸 것이고, 도 13의 b는 전원을 인가한 후의 사상 테이블 및 데이터 블록의 상태를 나타낸 것이다.

- <191> 도 13의 a에 도시한 바와 같이, 프로세서(500)로부터 요구되는 기록 연산에 따라 새로 생성된 사상정보에 의해 LBN(2)의 사상정보가 갱신되면, TC(135)의 LBN(2)에 새로운 사상정보가 기록된다.
- <192> 이와 같이, TC(135)의 LBN(2)에 새로운 사상정보가 기록되면, LBN(2)은 사용되고 있는 논리 주소이기 때문에 유효하지 않은 상태(invalid)로 지정되고, 해당되는 물리적 주소 '56'에 프로세서(500)로부터 입력되는 데이터가 기록되기 시작한다.
- <193> 만일, 이러한 과정에서 갑작스런 전원 중단이 발생한다면 데이터 기록은 중단되고, 전원 인가 후 초기화 과정이 수행된다.
- <194> 초기 과정에 따라 TC(135)가 독출되면, TC(135)의 LBN(2)는 도 13의 b에 도시된 바와 같이, 데이터 기록이 중단되었기 때문에 유효한 상태(valid)로 지정된다.
- <195> 플래시 메모리 컨트롤러(300)에서는 초기화 과정에서 독출된 PAT(131), BIT(133), TC(135) 또는 초기화 과정에서 독출된 PAT(131)와 최근 사상 정보가 삭제된 BIT(133), TC(135)를 UT(137)에 병합하고 UT(137)의 첫 자유블록(valid)을 검사하면 이전에 기록되던 물리 주소가 존재하는 것으로 검사된다.
- <196> 이에 따라, 플래시 메모리 컨트롤러(300)는 오류 블록을 복구하기 위하여, 첫 자유블록으로 지정된 PBN(2)에 기록된 물리 주소를 통해 해당되는 데이터 블록(150)에 기록된 데이터를 삭제한다.
- <197> 이러한 과정을 통해 갑작스런 전원 중단이 발생된 후, 전원이 인가되면 플래시 메모리 컨트롤러(300)에서는 프로세서(500)로부터 요구된 기록 또는 삭제 연산에 따른 재사상과정에서의 오류 및 데이터 기록 과정에서의 오류를 감지하고 복구할 수 있다.

<198> 본 발명은 도면에 도시된 일 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

【발명의 효과】

<199> 본 발명에 따르면, 플래시 메모리 액세스 시에 데이터 블록의 상태 정보를 포함하는 사상 테이블을 참조하여 읽기 또는 쓰기가 요구되는 데이터 블록을 효율적으로 액세스할 수 있다.

<200> 또한, 갑작스런 전원 중단으로 인해 발생하는 오류 블록을 감지하고 효과적으로 복구할 수 있는 효과를 제공한다.

<201> 그리고, 이러한 효과를 통해 플래시 메모리의 안정성을 극대화 시킬 수 있는 효과를 제공한다.

【특허청구범위】**【청구항 1】**

데이터 블록을 구성하는 각 블록에 할당되는 물리 주소와 각 블록의 상태 정보로 이루어지는 사상 테이블인 제 1 사상 테이블과,

오류 블록이 제외된 상기 제 1 사상 테이블의 각 블록에 대한 물리 주소와 논리 주소간의 사상정보로 이루어지는 사상테이블인 제 2 사상 테이블과,

상기 제 2 사상 테이블의 갱신동작을 최소화하기 위해 최근의 사상정보를 지정된 값만큼 기록하고 처리하는 사상테이블인 제 3 사상 테이블로 이루어진 맵 블록을 포함하는 것을 특징으로 하는 플래시 메모리.

【청구항 2】

제 1항에 있어서,

상기 제 1 사상 테이블, 제 2 사상 테이블, 제 3 사상 테이블에 대한 각각의 여유 블록을 더 포함하는 것을 특징으로 하는 플래시 메모리.

【청구항 3】

제 2항에 있어서,

상기 제 2 사상 테이블 및 제 3 사상 테이블에 대한 여유 블록은 기록 및 삭제 연산에 따라 발생하는 사상정보로 인해 갱신되는 이전의 사상정보를 저장하는 것을 특징으로 하는 플래시 메모리.

【청구항 4】

데이터 블록의 물리 주소 정보로 이루어지는 제 1 사상 테이블과 오류블록이 제외된 제 1 사상 테이블의 사상정보로 이루어지는 제 2 사상 테이블과 가장 최근의 사상정보로 이루어지는 제 3 사상 테이블과 각각의 여유 블록으로 이루어지는 맵 블록을 포함하는 플래시 메모리,

상기 플래시 메모리의 맵 블록으로부터 얻어지는 제 1 및 제 2 및 제 3 사상 테이블을 통해 자유블록 정보로 이루어지는 제 4 사상 테이블을 생성하고 판독 및 기록 연산시 제 2 및 제 3 사상 테이블 또는 제 4 사상 테이블을 참조하여 데이터를 판독/기록할 물리 주소를 액세스하는 플래시 메모리 컨트롤러를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

【청구항 5】

제 4항에 있어서,

상기 플래시 메모리 컨트롤러는 상기 제 2 사상 테이블과 제 3 사상 테이블 및 제 4 사상 테이블을 통해 사상정보 갱신 또는 데이터 기록 과정에서의 전원 중단에 의한 오류를 감지하고 데이터를 복구하는 것을 특징으로 하는 플래시 메모리 액세스 장치.

【청구항 6】

프로세서로부터 판독 및 기록 연산이 요구되면 커맨드와 함께 논리 주소를 입력받는 단계;

상기 판독 및 기록 연산을 위해 플래시 메모리로부터 독출되는 데이터 블록의 물리 주소 정보 가운데 오류블록이 제외된 사상정보로 이루어지는 제 2 사상 테이블과 가

장 최근의 사상정보로 이루어지는 제 3 사상 테이블에서 상기 논리 주소를 체크하는 단계;

상기 제 2 사상 테이블과 제 3 사상 테이블에 상기 논리 주소가 존재하면 지정되는 데이터 블록의 물리 주소를 액세스 하여 판독 및 기록 연산을 처리하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 7】

제 6항에 있어서,

상기 제 2 사상 테이블과 제 3 사상 테이블에 상기 논리 주소가 존재하지 않으면 판독 또는 기록 연산에 따라 판독 오류 처리하거나 플래시 메모리 컨트롤러에 의해 생성되는 자유블록 정보로 이루어지는 제 4 사상 테이블을 통해 할당되는 자유블록의 물리 주소를 액세스하여 기록 연산을 처리하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 8】

제 6항에 있어서,

상기 프로세서로부터 요구되는 연산 동작에 따라 상기 플래시 메모리를 효율적으로 액세스하기 위한 사상정보를 플래시 메모리의 맵 블록으로부터 로딩하여 플래시 메모리를 초기화하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 9】

제 6항에 있어서,

상기 기록 연산 과정에서 발생하는 전원 중단에 의한 오류를 감지하고 데이터를 복구하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 10】

제 6항에 있어서,

상기 판독 연산은

상기 프로세서로부터 판독 연산이 요구되면 데이터 판독을 위한 소정의 논리 주소를 입력받는 단계;

상기 입력되는 논리 주소를 상기 제 2 사상 테이블과 제 3 사상 테이블을 참조하여 실제로 데이터가 기록된 플래시 메모리의 물리 주소로 변환하는 단계;

상기 물리주소로의 변환 과정에서 프로세서로부터 판독이 요구되는 논리 주소가 데이터 블록에 존재하는 유효한 주소인지를 판별하는 단계;

상기 판별결과 유효한 주소가 아닌 경우에는 판독 오류로 처리하고 유효한 주소인 경우에는 상기 물리 주소를 통해 플래시 메모리에 기록된 데이터를 독출하여 상기 프로세서로 전송하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 11】

제 6항에 있어서,

상기 기록 연산은

상기 프로세서로부터 기록 연산이 요구되면 데이터 기록을 위한 소정의 논리 주소를 입력받는 단계;

상기 입력된 논리 주소가 상기 제 2 및 제 3 사상 테이블에 존재하는지 여부를 판별하는 단계;

상기 판별결과에 따라 단계적 사상기법을 통해 지정되는 임의의 블록 또는 플래시 메모리 컨트롤러에 의해 생성되는 자유블록 정보로 이루어지는 제 4 사상 테이블을 통해 할당되는 자유블록을 데이터가 기록될 물리 주소로 지정하는 단계;

상기 지정된 물리 주소에 따라 상기 논리 주소와 물리 주소간의 사상정보를 상기 제 3 사상 테이블에 기록하고 갱신되는 사상정보에 따라 사용 영역을 나타내는 상기 제 3 사상 테이블의 포인터를 변경하는 단계;

상기 프로세서로부터 전송되는 데이터를 버퍼로 입력받고 입력된 데이터를 상기 물리 주소에 기록하는 단계;

상기 데이터 기록 과정에서의 오류 발생 여부를 판별하는 단계;

상기 판별결과 오류가 발생되면 플래시 메모리로부터 독출되는 데이터 블록의 물리 주소 정보로 이루어지는 제 1 사상 테이블에 상기 물리 주소의 블록을 오류블록으로 기록하고 갱신되는 오류정보에 따라 현재의 사상 정보를 나타내는 주소정보를 갱신된 제 1 사상 테이블이 기록된 주소로 변경하며 자유블록을 검색하여 데이터를 기록을 시도하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 12】

제 11항에 있어서,

상기 플래시 메모리에 할당된 제 3 사상 테이블의 사상정보 저장 용량이 이전의 사상정보로 인해 부족하면, 상기 제 3 사상 테이블의 사상정보를 상기 제 2 사상 테이블

로 갱신하기 위해 새로운 블록을 할당받아 상기 제 3 사상 테이블의 사상 정보가 갱신되는 제 2 사상 테이블의 사상정보를 할당된 블록에 기록하는 단계;

상기 제 3 사상 테이블에 기록된 이전의 사상 정보를 삭제하는 단계;

상기 갱신 또는 삭제되는 사상정보에 따라 사용 영역을 나타내는 상기 제 2 및 제 3 사상 테이블의 포인터를 변경하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 13】

제 11항에 있어서,

상기 플래시 메모리에 할당된 제 2 및 제 3 사상 테이블의 사상정보 저장 용량이 이전의 사상정보로 인해 부족하면, 상기 제 3 사상 테이블의 사상정보를 상기 제 2 사상 테이블로 갱신하기 위해 새로운 블록을 할당받아 상기 제 3 사상 테이블의 사상 정보가 갱신되는 제 2 사상 테이블의 사상정보를 할당된 블록에 기록하는 단계;

상기 제 2 및 제 3 사상 테이블에 기록된 이전의 사상 정보를 삭제하는 단계;

상기 갱신 또는 삭제되는 사상정보에 따라 사용 영역을 나타내는 상기 제 2 및 제 3 사상 테이블의 포인터를 변경하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 14】

제 11항에 있어서,

상기 논리 주소에 기존에 기록된 데이터가 존재하면, 상기 프로세서로부터 입력되는 데이터를 버퍼로 입력받고, 입력된 데이터와 기존에 기록된 데이터를 병합하여 병합된 데이터를 상기 물리 주소에 기록하는 단계;

상기 데이터 기록이 완료되면 기존에 데이터가 기록되었던 블록의 데이터를 삭제하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 15】

제 11 내지 13항 중 어느 한 항에 있어서,

상기 제 2 및 제 3 사상 테이블의 사상 정보를 갱신할 때마다 각 사상 테이블의 여유 블록에 갱신되는 이전의 사상정보를 저장하는 단계를 더 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 16】

제 8항에 있어서,

상기 플래시 메모리를 초기화 하는 단계는

상기 플래시 메모리의 지정된 블록에 기록되는 제 1 사상 테이블 주소정보에 대한 저장정보를 포함하는 플래시 메모리에 대한 전체 정보를 체크하는 단계;

상기 플래시 메모리의 맵 블록으로부터 제 1 사상 테이블을 독출하고, 맵 블록에 할당된 제 2 및 제 3 사상 테이블 영역을 통해 제 2 및 제 3 사상 테이블을 독출하는 단계;

상기 독출된 제 1 및 제 2 및 제 3 사상 테이블을 논리합으로 병합하여 제 4 사상 테이블을 생성하고, 생성된 제 4 사상 테이블을 램에 저장하는 단계;

상기 프로세서로부터 요구되는 판독 또는 기록 연산을 처리하기 위해 대기하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 17】

제 16항에 있어서,

상기 제 1 사상 테이블을 독출하는 단계는

상기 저장정보를 이용하여 지정된 블록에서 제 1 사상 테이블의 주소정보를 검색하는 단계;

상기 검색결과 상기 주소정보가 검색되면 주소정보를 검출하는 단계;

상기 검색결과 상기 주소정보가 검색되지 않으면 상기 플래시 메모리에 할당되는 제 1 사상 테이블 블록 영역에서 상기 제 1 사상 테이블을 검색하여 상기 제 1 사상 테이블이 검색된 주소정보를 지정된 블록에 기록하거나 상기 제 1 사상 테이블 블록 영역을 검사하여 제 1 사상 테이블을 생성하고 생성된 제 1 사상 테이블이 기록된 주소정보를 지정된 블록에 기록하는 단계;

상기 검색 또는 생성된 주소정보를 이용하여 상기 플래시 메모리로부터 제 1 사상 테이블을 독출하는 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

【청구항 18】

제 9항에 있어서,

상기 오류를 감지하고 데이터를 복구하는 단계는

상기 제 2 및 제 3 사상 테이블을 상기 플래시 메모리의 맵 블록에서 체크하여, 현재 사용되는 제 2 및 제 3 사상 테이블이 복수개 존재하는지 여부를 판별하는 단계;

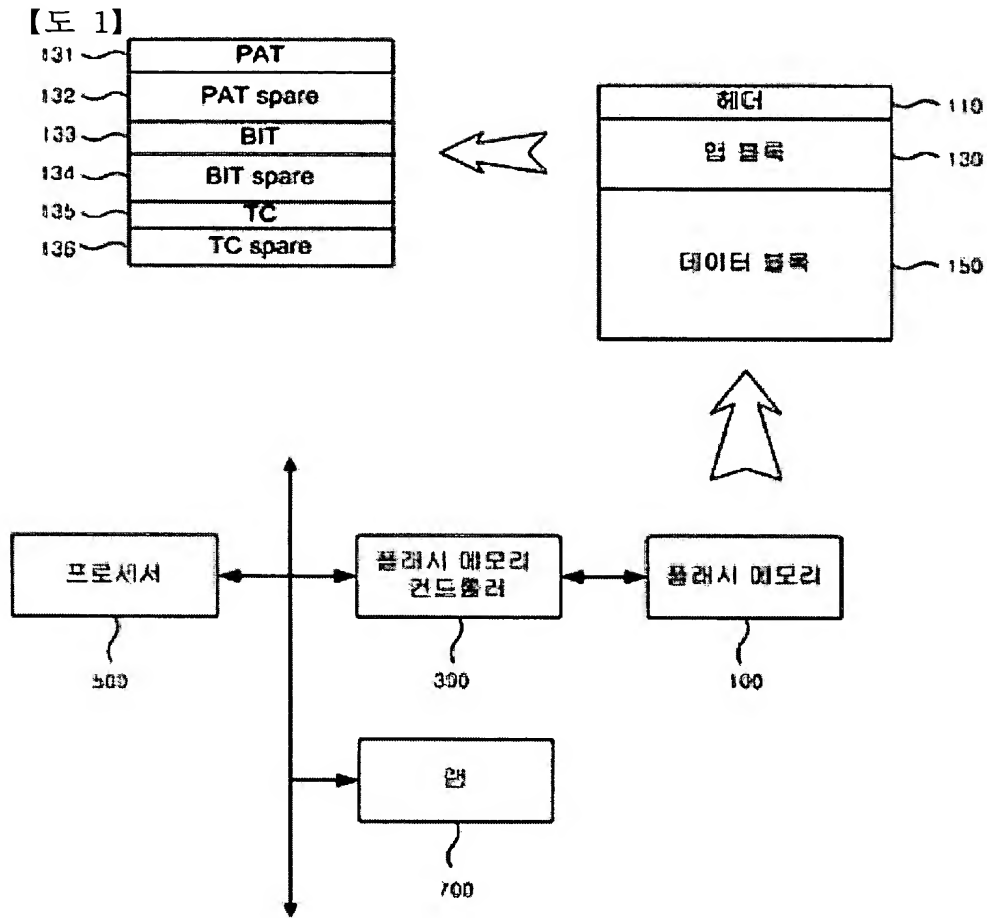
상기 판별결과 제 2 및 제 3 사상 테이블이 복수개 존재하면 사상정보를 갱신하는 과정에서 전원 중단에 의해 오류가 발생된 것으로 판단하여 상기 제 2 및 제 3 사상 테이블에 기록된 가장 최근의 사상정보를 삭제하는 단계;

상기 판별결과에 따라 검색된 제 2 및 제 3 사상 테이블 또는 최근의 사상 정보가 삭제된 제 2 및 제 3 사상 테이블을 상기 플래시 메모리로부터 독출되는 제 1 사상 테이블과 함께 초기화 과정에서 플래시 메모리 컨트롤러에 의해 생성되는 제 4 사상 테이블에 병합시키는 단계;

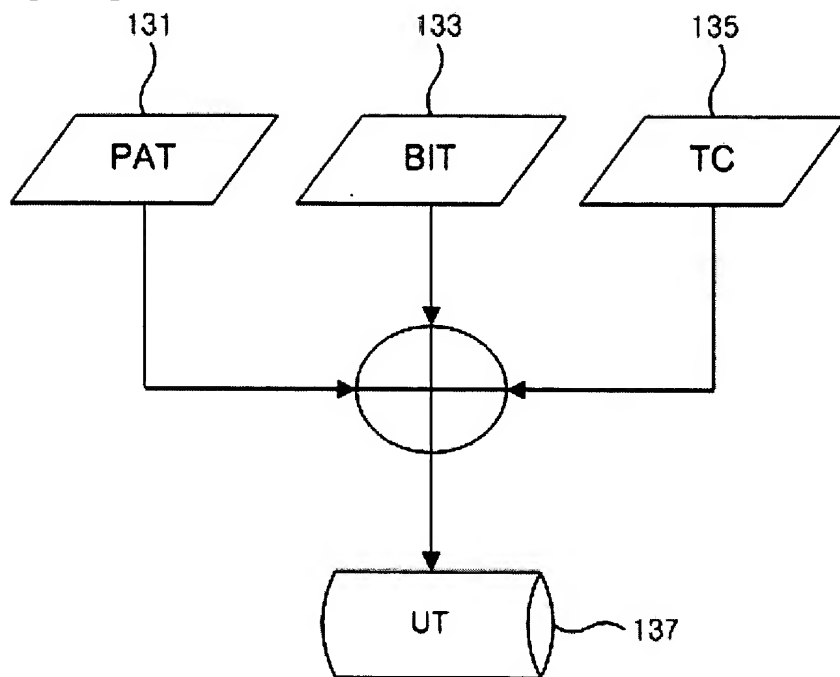
상기 병합된 제 4 사상 테이블에서 첫 자유블록을 검색하여 검색된 자유블록이 기록된 데이터가 없는 순수한 자유블록인지 여부를 판별하는 단계;

상기 판별결과 순수한 자유블록이 아닌 경우에는 데이터를 기록하는 과정에서 전원 중단에 의해 오류가 발생된 것으로 판단하여 기록된 데이터를 삭제하는 단계를 단계를 포함하는 것을 특징으로 하는 플래시 메모리 액세스 방법.

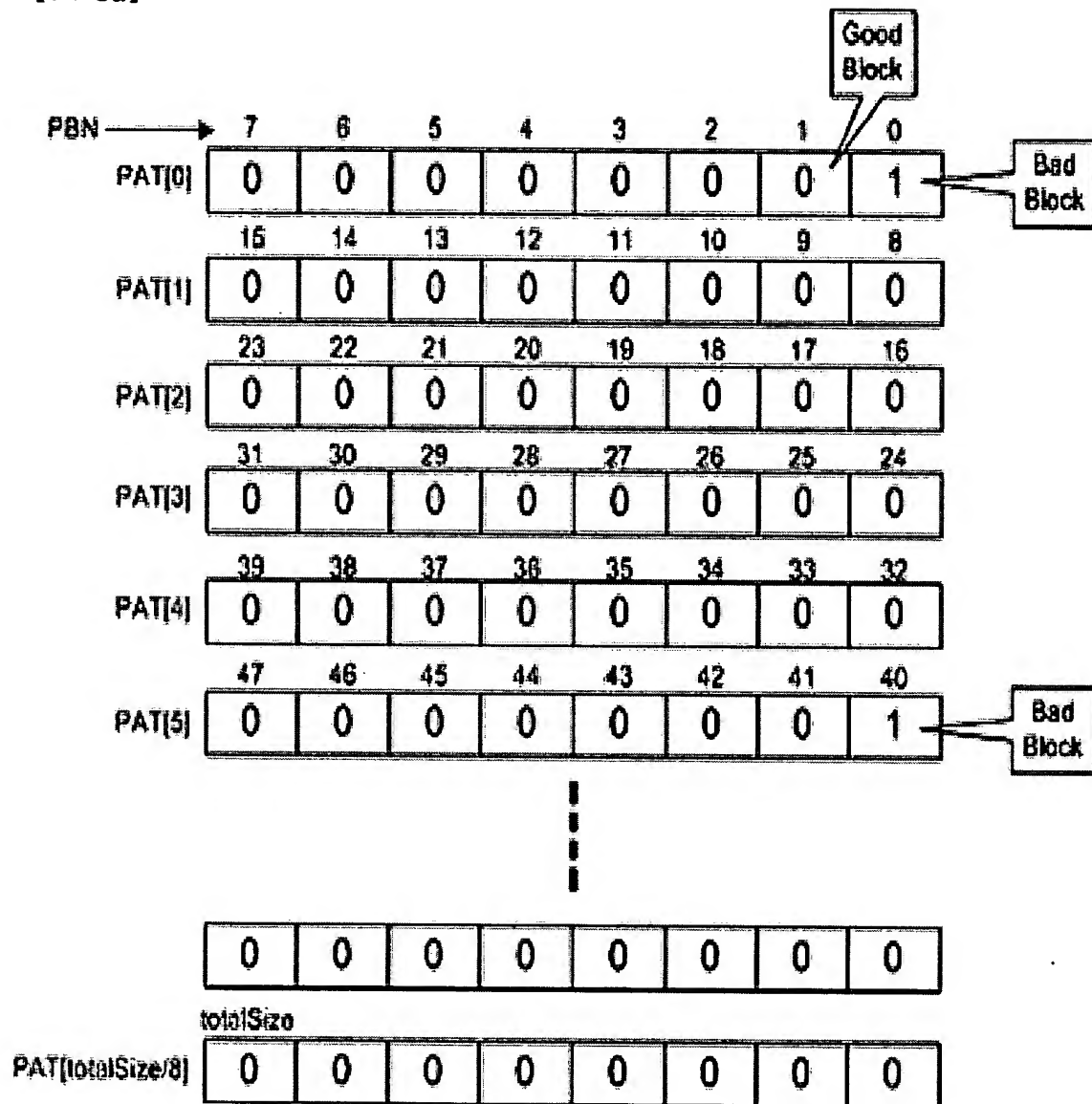
【도면】



【도 2】



【도 3a】



【도 3b】

개선전

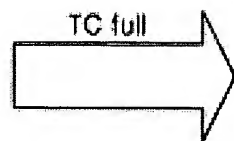
BIT

LBN	PBN
0	0xFFFF
1	0xFFFF
2	0xFFFF
3	0xFFFF
4	0xFFFF

⋮

TC

LBN	PBN
0	1
1	4
2	56

개선후

BIT

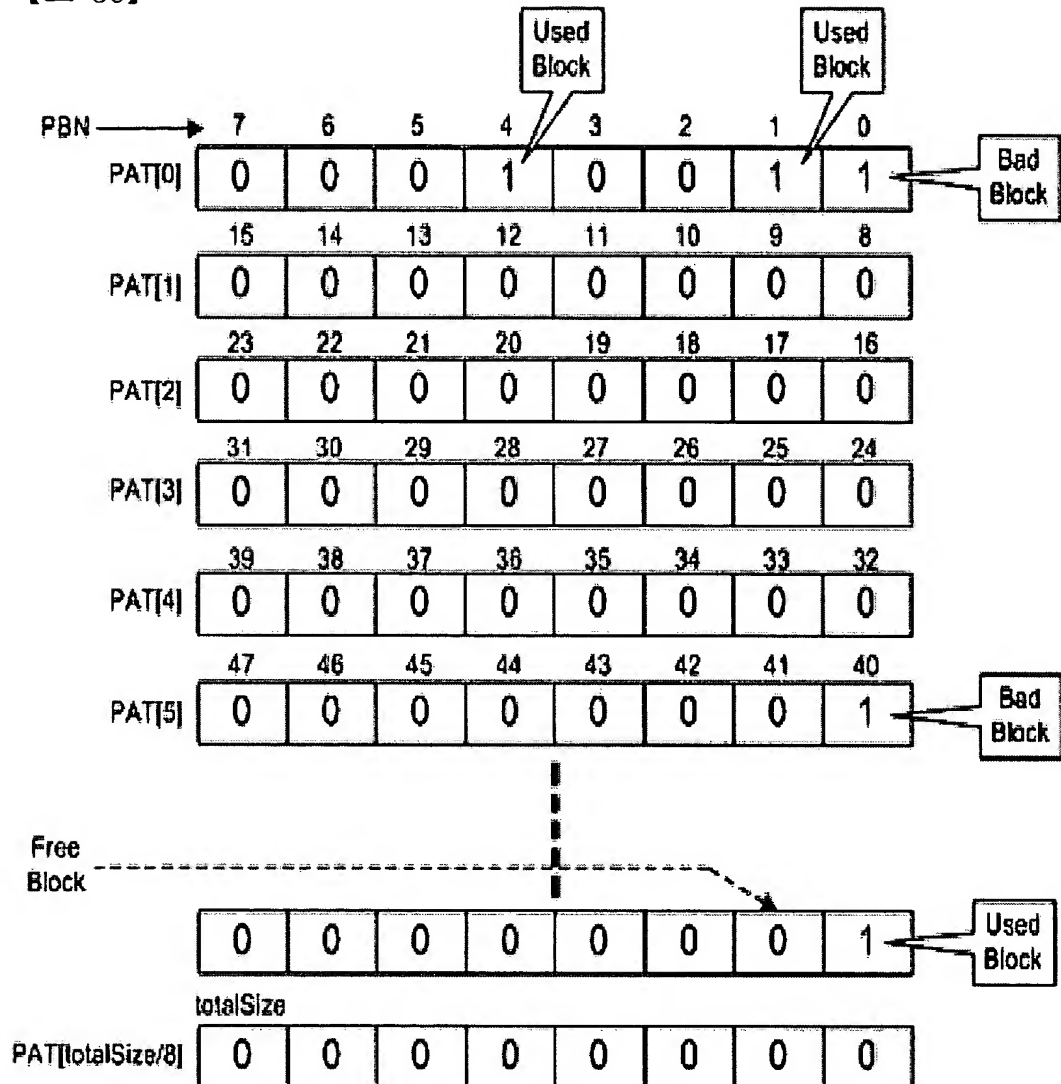
LBN	PBN
0	1
1	4
2	56
3	0xFFFF
4	0xFFFF

⋮

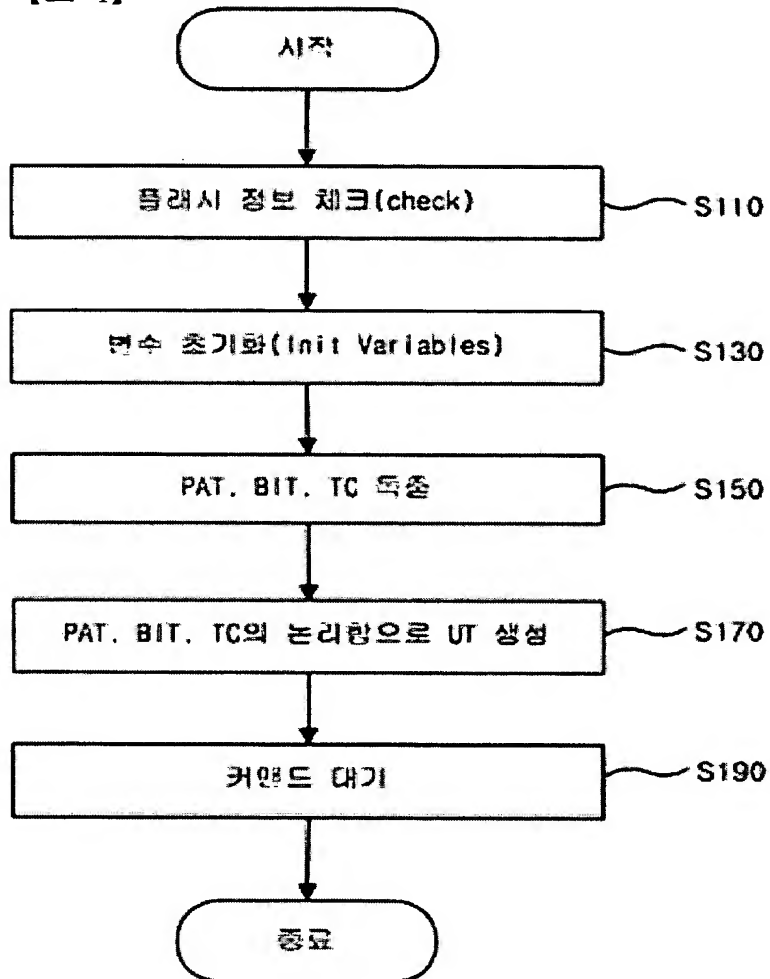
TC

LBN	PBN
0	0xFFFF
1	0xFFFF
2	0xFFFF

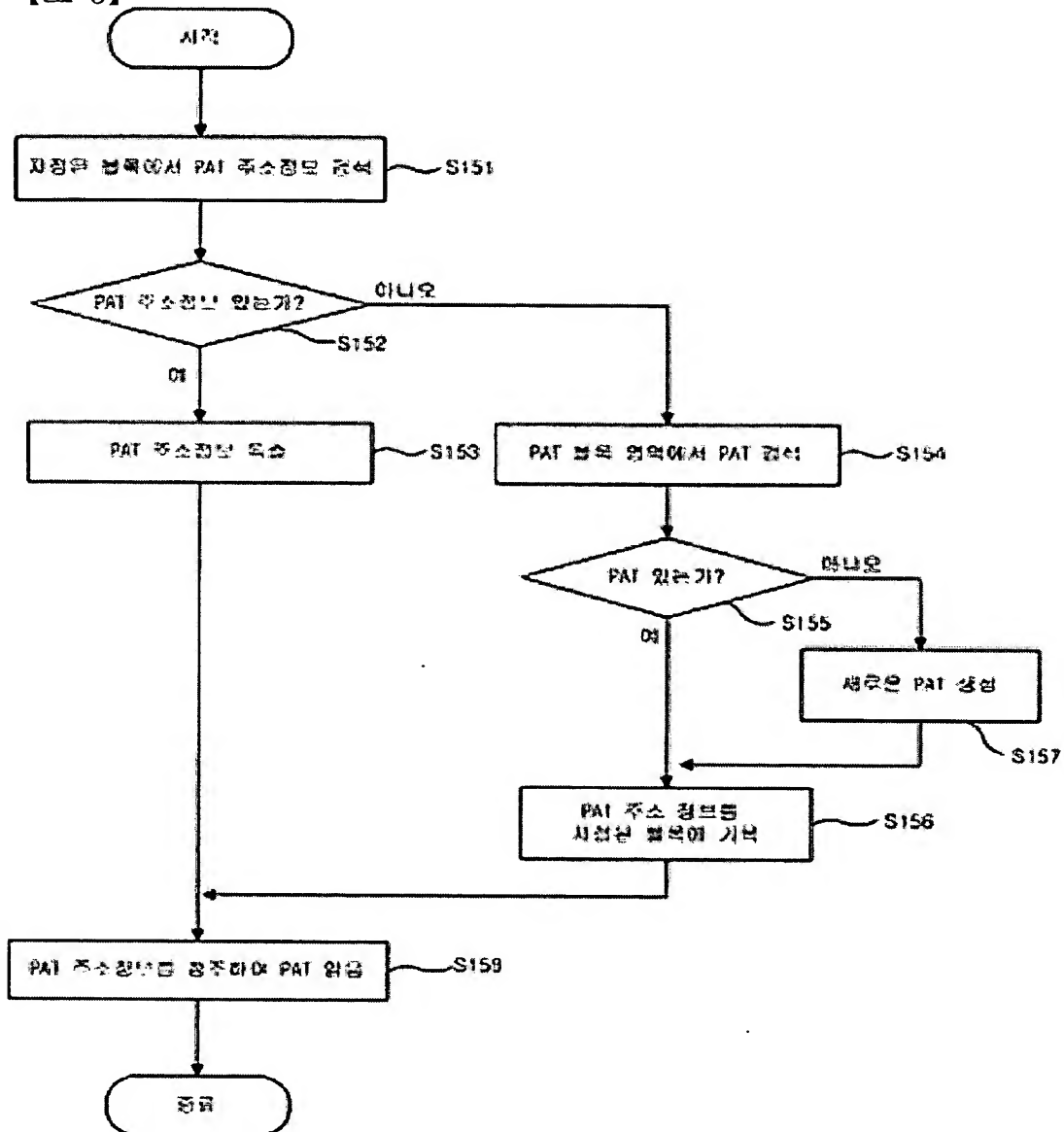
【도 3c】



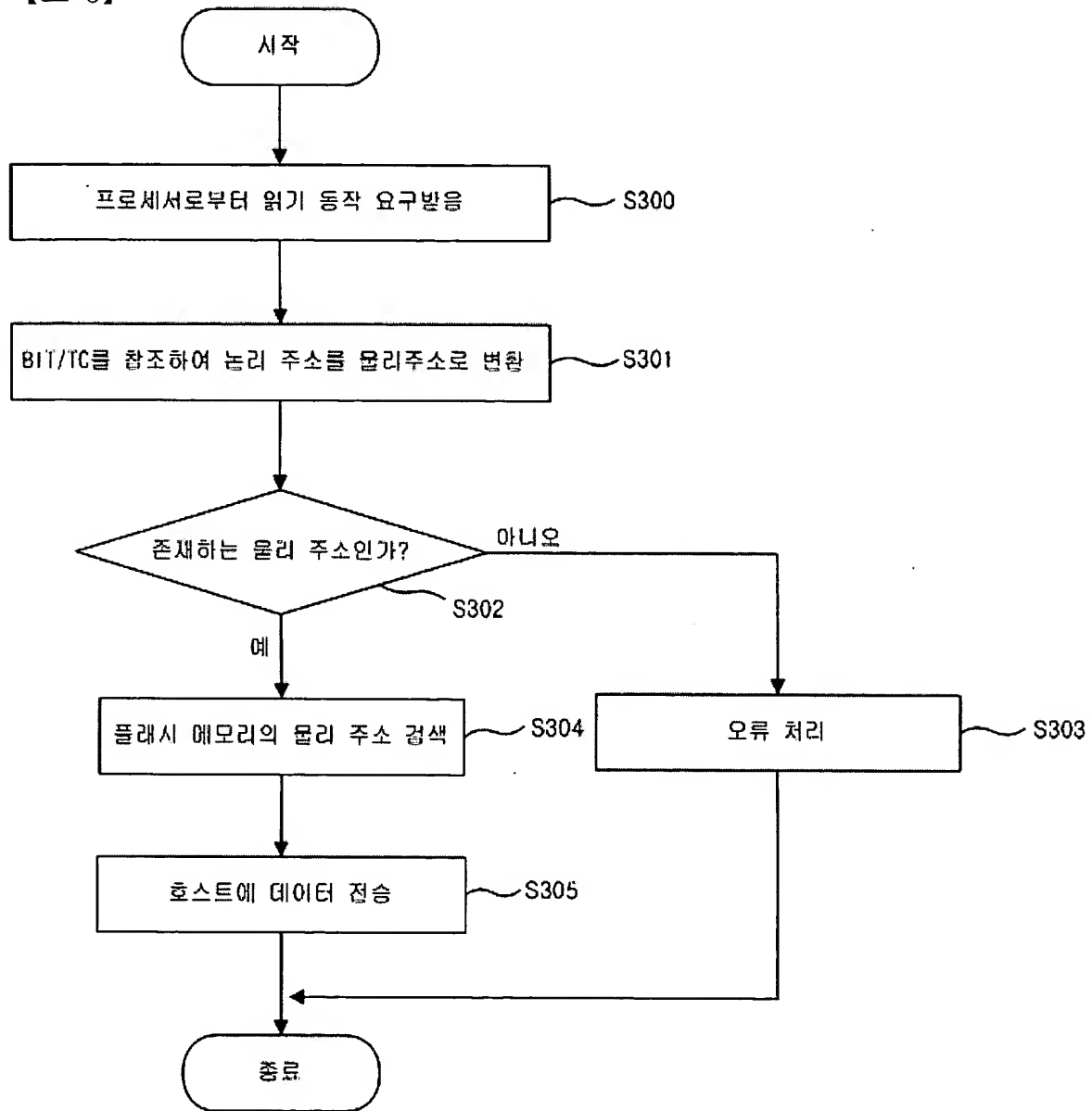
【도 4】



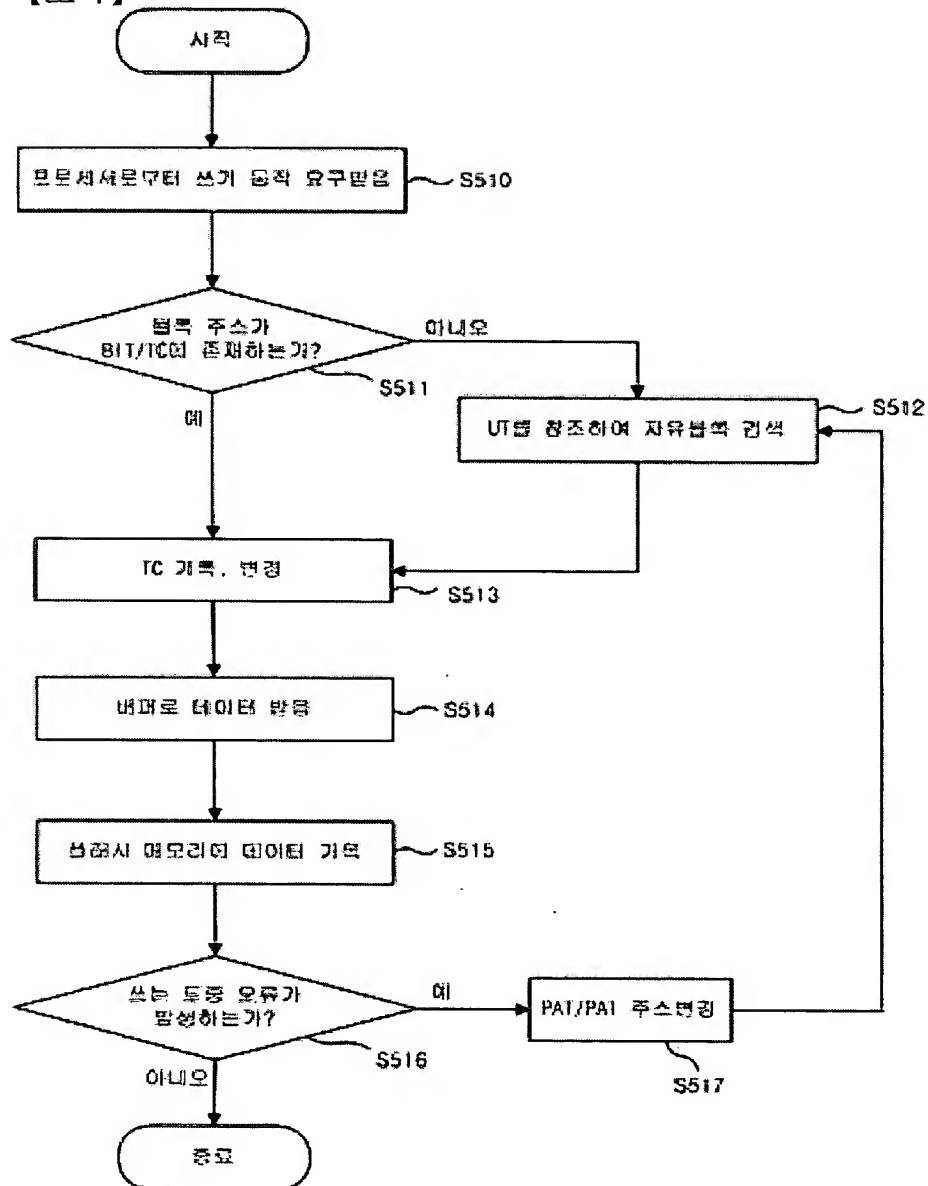
【도 5】



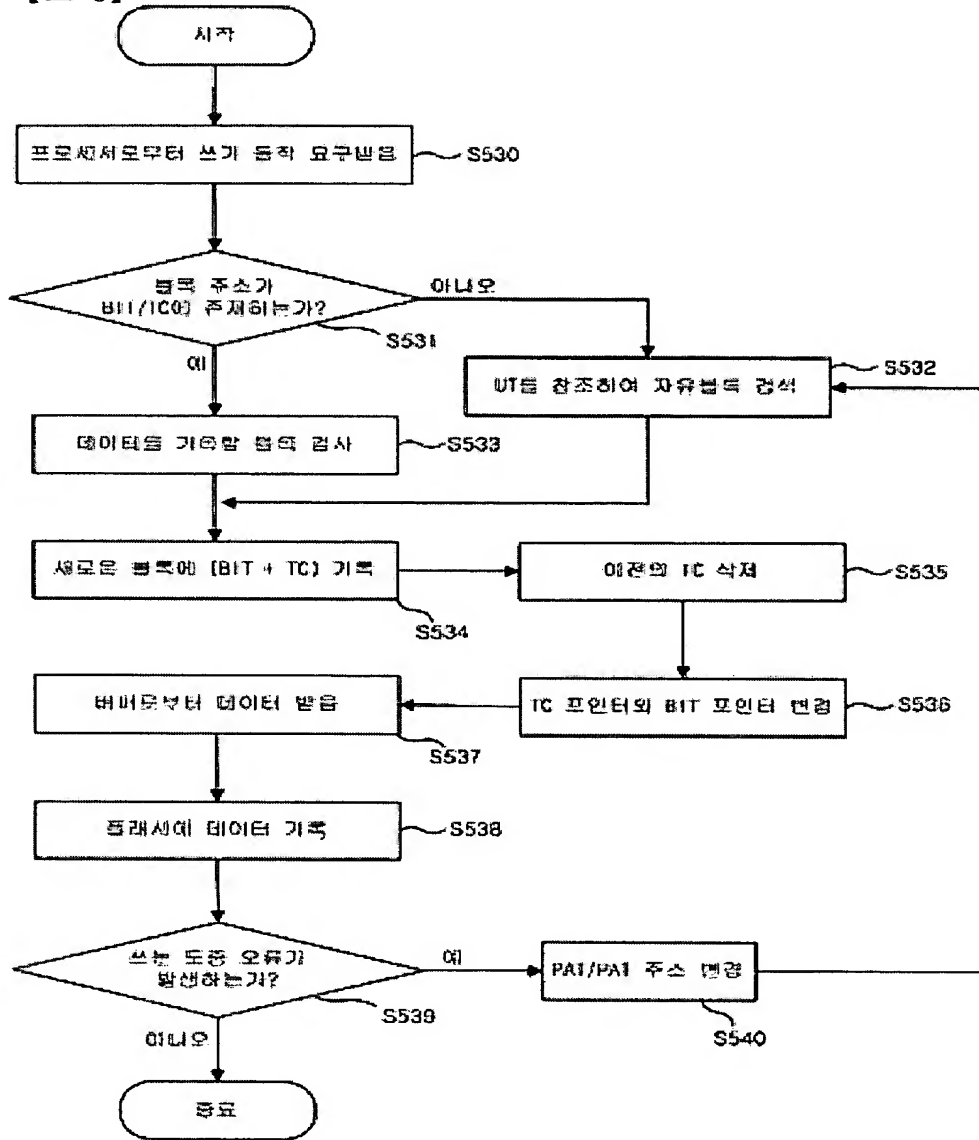
【도 6】



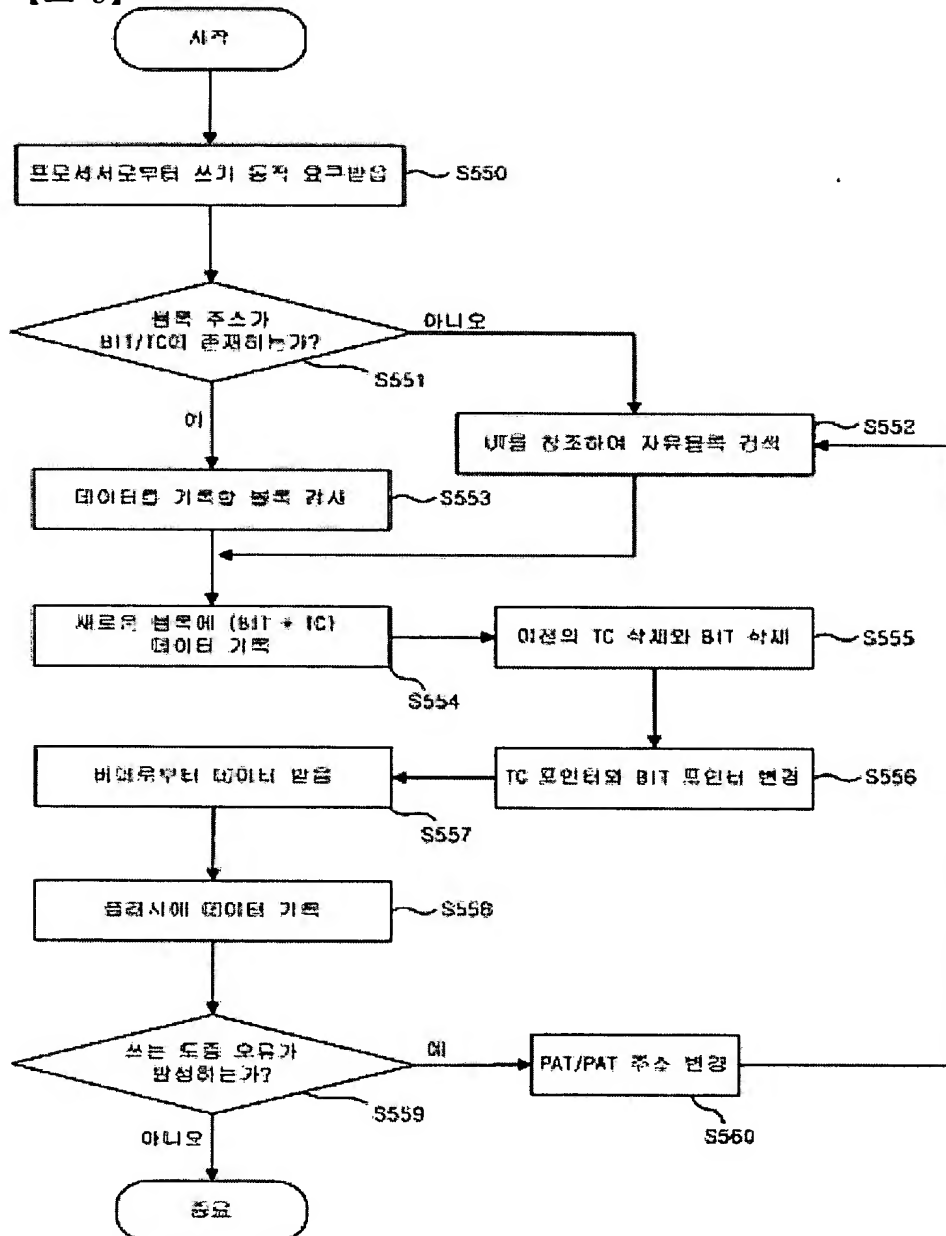
【도 7】



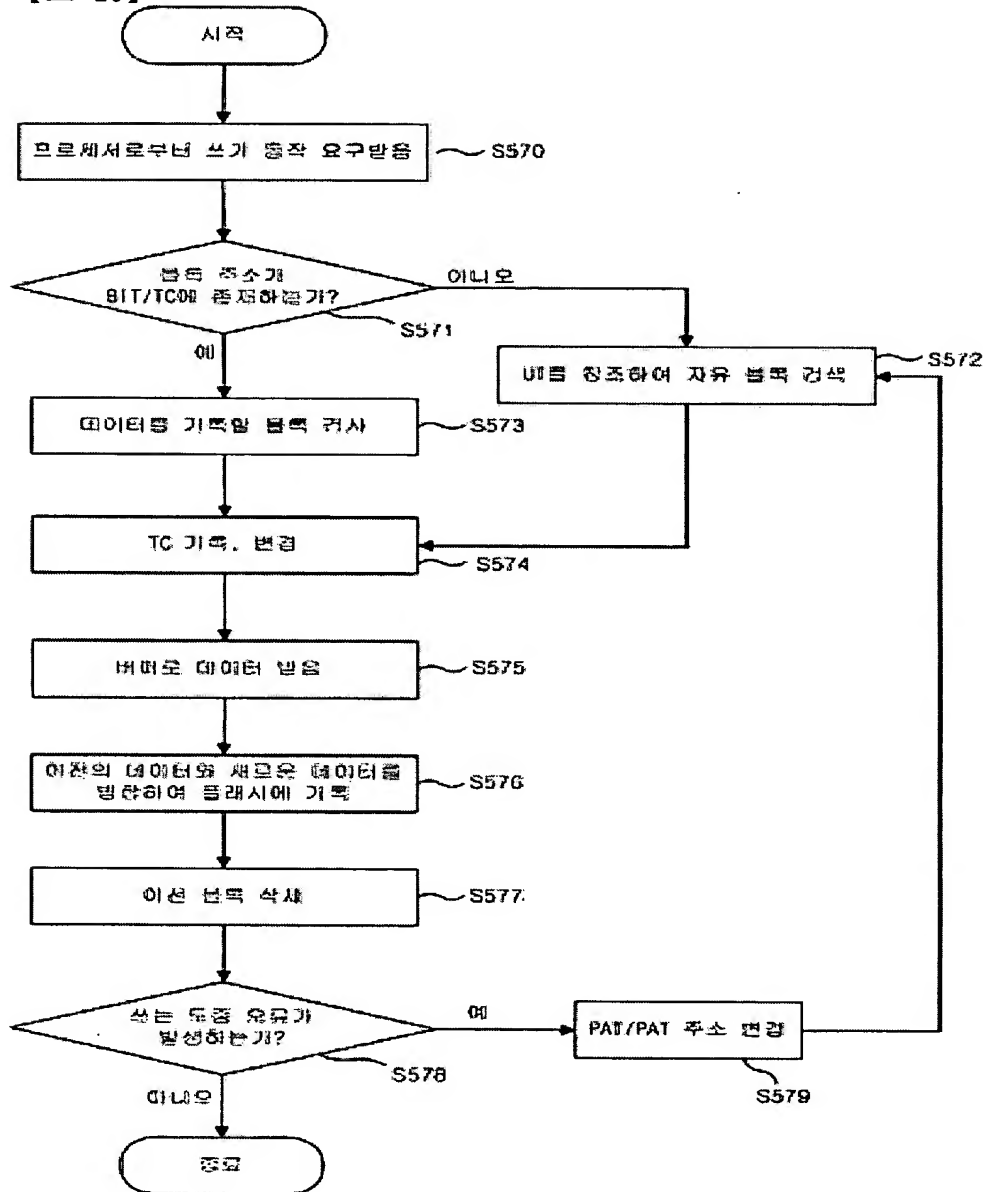
【도 8】



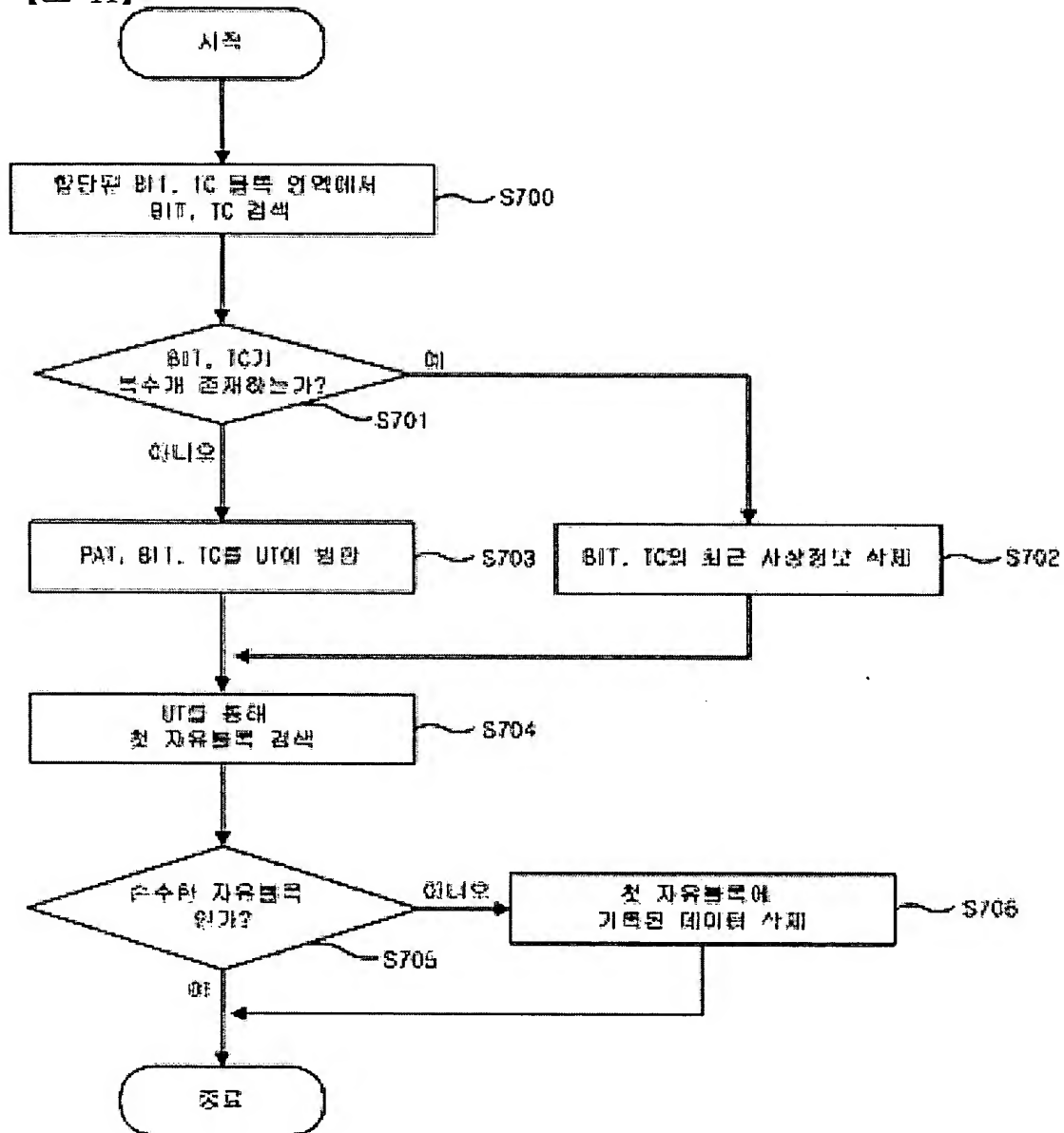
【도 9】



【도 10】



【도 11】



【도 12a】

TC

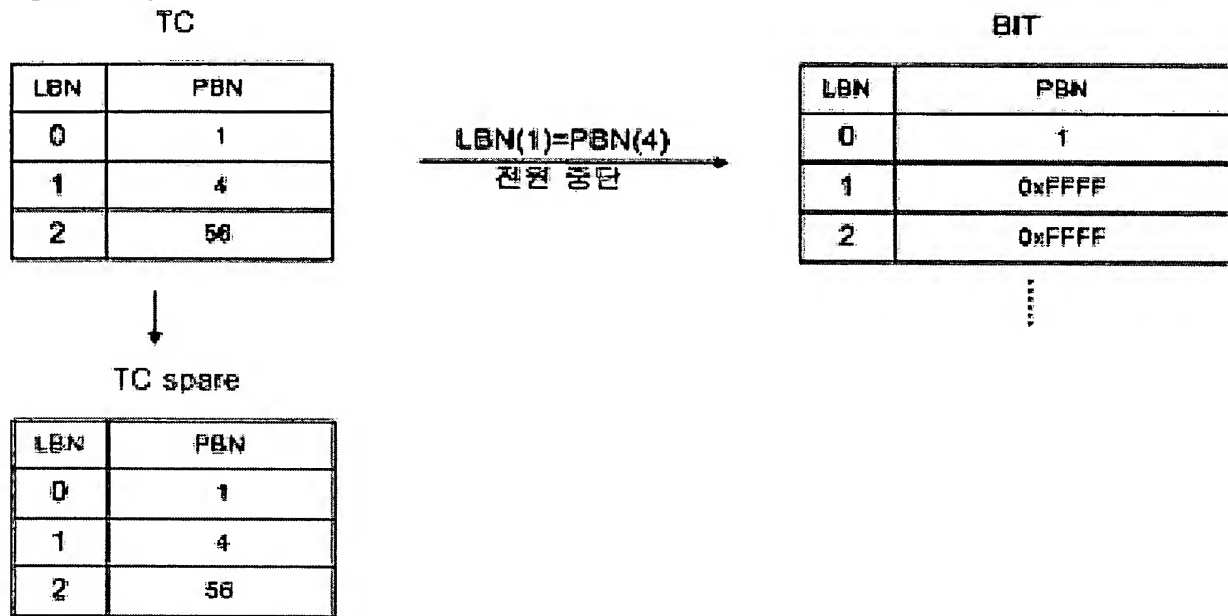
LBN	PBN
0	1
1	4
2	0xFFFF

BIT

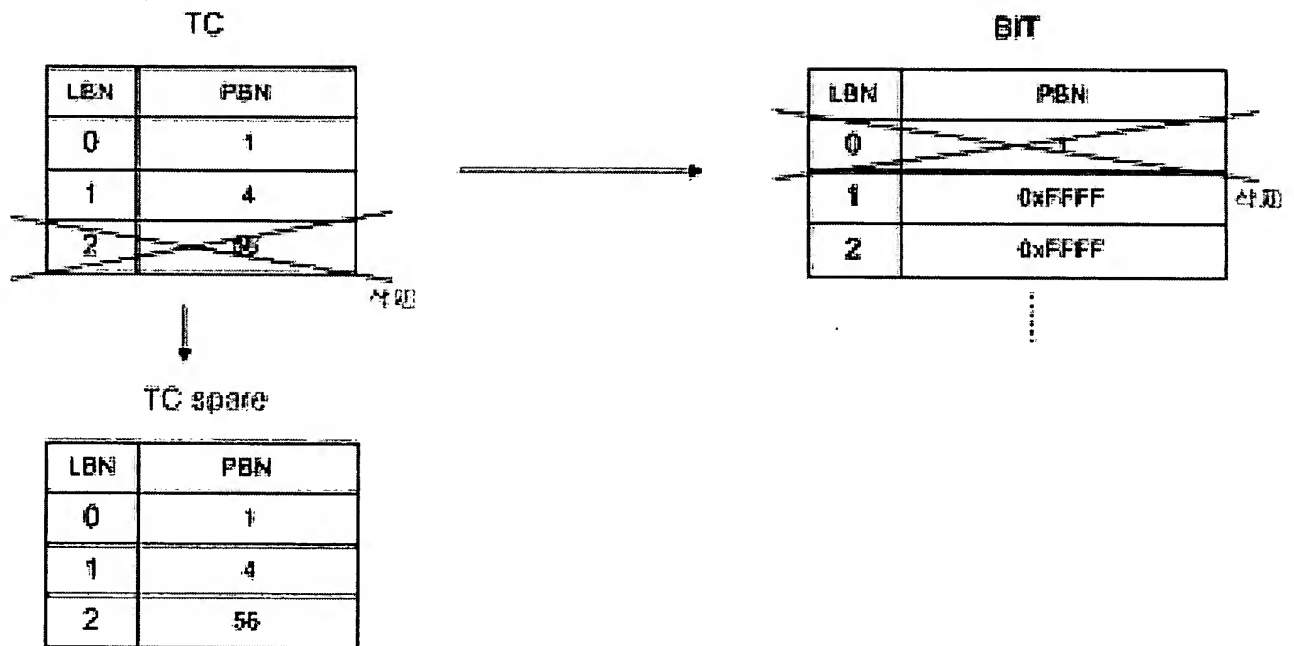
LBN	PBN
0	0xFFFF
1	0xFFFF
2	0xFFFF

⋮

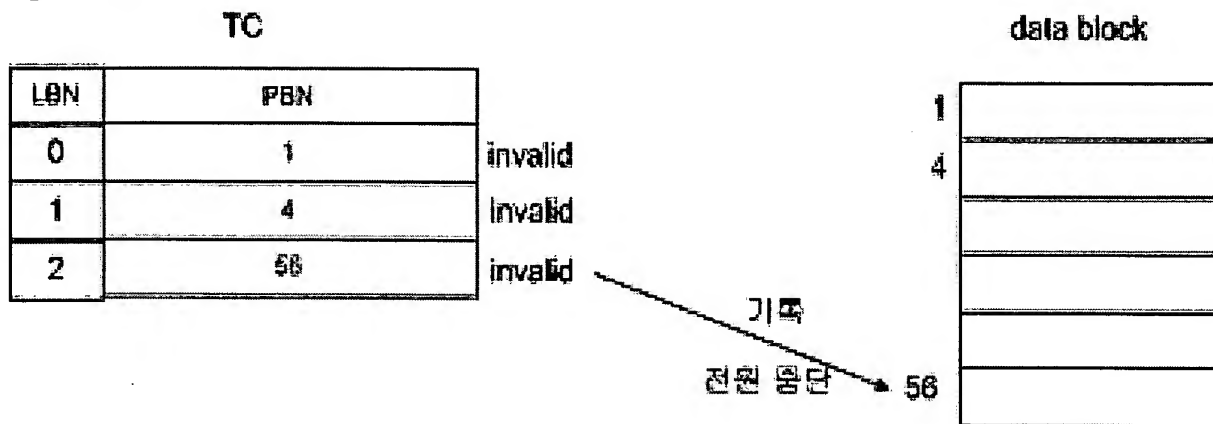
【도 12b】



【도 12c】



【도 13a】



【도 13b】

